

# **Spotfire® Web Client User Guide**

*Software Release 14.4.1*

# Contents

---

<b>This is Spotfire.....</b>	<b>13</b>
The user interface.....	14
Load data.....	15
Know the data.....	16
Visualize data.....	19
Analyze data.....	23
Make calculations on the data.....	26
Share with others.....	28
Spotfire clients and user roles.....	29
Logging in to the installed Spotfire client.....	30
Logging in to the web client.....	37
<b>Loading data.....</b>	<b>38</b>
Choosing the data to analyze.....	39
Creating a new analysis from data in the library.....	40
Creating a new analysis from a local file.....	41
Linked copy to data table in analysis.....	43
Types of data in Spotfire.....	45
Choosing how to load the data.....	55
Load methods.....	58
Linked, stored, and embedded data.....	70
Reloading data for each user when using scheduled updates.....	71
Adding more data.....	73
Related data tables, joins and column matches.....	74
Data from databases.....	81
Accessing data from a database.....	82
Accessing data with a shared connection or information link.....	85
Supported data sources.....	86
Data connections and connection data sources.....	109
Data connections in the Spotfire web client.....	113
Logging in to a data connection.....	114
Properties in connection data sources.....	115
Configuring other driver settings as custom properties.....	116
Single sign-on with an identity provider (OAuth2) for connectors.....	118
Connecting to Data from TIBCO Data Virtualization (TDV).....	120
Data preparation in Microsoft Excel.....	121

Searching the library.....	124
Creating an RSS feed of your Spotfire Library.....	131
Using search expressions in shareable links.....	131
Opening an analysis saved in the library.....	132
Limiting the data to load.....	132
<b>Handling data.....</b>	<b>135</b>
Data in analysis.....	135
The expanded data in analysis flyout.....	136
Changing the categorization of a column.....	140
Changing the formatting of a column.....	142
Changing the default sort order of values in a column.....	156
Changing the data type of a column.....	158
Changing a column name.....	158
Setting display values based on other columns.....	159
Splitting a column.....	165
Deleting a column.....	168
Geographic location and geocoding.....	169
Unpivoting columns.....	174
Data canvas.....	178
Changing the name of a data table.....	181
Deleting a data table.....	182
Adding rows to a data table.....	182
Adding columns to a data table.....	187
Storing data within the analysis.....	193
Removing operations in the data canvas.....	194
Removing a data source from a data table.....	195
Editing a data connection.....	197
Specifying key columns for a data table.....	199
Data tables overview.....	201
Data tables in Spotfire.....	204
Multiple data tables in one visualization.....	206
Replacing data.....	226
Reloading data.....	227
Reloading data for a single data source.....	228
Replacing a data table.....	230
Replacing a data source.....	233
Replacing empty values in a column.....	237
Replacing a value.....	239
Document properties, column properties and data table properties.....	243

Adding or editing a document property.....	245
Adding a new data table or column property.....	246
Using document, data table or column properties in an analysis.....	250
Understanding properties in a Spotfire analysis.....	273
<b>Visualizations.....</b>	<b>277</b>
Creating a visualization.....	277
Table.....	280
Creating a table.....	281
Changing the order of the columns in a table.....	282
Hiding a column in a table.....	282
Adding columns to a table.....	283
Changing height and width in tables.....	283
Sorting a table.....	284
Specifying the header orientation and header text alignment.....	286
Specifying the text alignment for table cells.....	287
Specifying text color, background color, and font style in a table.....	288
Showing text as hyperlinks.....	288
Showing images in tables.....	288
Virtual columns.....	289
Cross table.....	290
Creating a cross table.....	292
Displaying totals.....	294
Changing height and width in tables.....	298
Sorting values in a cross table.....	299
Styling in the cross table.....	302
Formatting in the cross table.....	307
Marking values in cross tables.....	310
Images in the cross table.....	311
Graphical table.....	312
Creating a graphical table.....	313
Adding dynamic items to a graphical table.....	314
Removing a dynamic item from a graphical table.....	315
Changing the column header for a dynamic item.....	316
Hiding the column header for a dynamic item.....	316
Changing height and width in tables.....	317
Sorting the rows in a graphical table.....	318
Copying marked rows in a graphical table.....	318
Making a column not respond to data limiting.....	318
Adding or editing a rule for the color, shape, or font style of a dynamic item (icons, calculated values).....	319



Sparklines.....	320
Calculated values.....	321
Icons.....	322
Bullet graphs.....	322
Text area.....	324
Adding a property control to a text area.....	325
Adding a dynamic item to a text area.....	327
Changing the size of a dynamic item in a text area.....	328
Changing the font, style and size of calculated values in a text area.....	329
Edit text area toolbar.....	329
Bar chart.....	331
Creating a bar chart.....	333
Creating a stacked bar chart.....	334
Creating a side-by-side bar chart.....	338
Creating a 100% stacked bar chart.....	341
Creating a horizontal bar chart.....	344
Creating a histogram.....	346
Showing values of bars.....	349
Sorting bars by value.....	350
Sorting bar segments by value.....	351
Showing only top bars.....	352
Waterfall chart.....	355
Creating a waterfall chart.....	356
Showing intermediate total bars.....	359
Showing values of bars and blocks.....	363
Showing waterfall per categorical color.....	364
Showing waterfall per category on the category axis.....	365
Continuing a waterfall over trellis panels.....	366
Line chart.....	367
Creating a line chart.....	368
Marking in line charts.....	370
Combination chart.....	371
Creating a combination chart.....	373
Sorting the combination chart.....	377
Switching between bars and line series.....	378
Pie chart.....	379
Creating a pie chart.....	380
Sorting pie sectors by size.....	381
Scatter plot.....	382
Creating a scatter plot.....	383

Using marker appearance to add more information.....	385
Using tiled markers.....	387
Creating a density plot.....	388
Adjusting the drawing order of markers.....	390
Showing directions using marker rotation in a scatter plot.....	391
Map chart.....	392
Creating a map chart.....	392
Working with layers.....	416
Viewing the coordinates for a position.....	418
Navigation in the map chart.....	419
Fixing geocoding issues.....	420
Aggregating coordinate-based locations values.....	421
Example of a line connection in a marker layer.....	423
Treemap.....	424
Creating a treemap.....	426
Navigating in a treemap hierarchy.....	428
Heat map.....	430
KPI chart.....	434
Creating a KPI chart.....	436
Adding a new KPI to a KPI chart.....	436
Specifying generic KPI chart properties.....	447
Showing sparklines.....	448
Sorting tiles alphabetically.....	450
Sorting tiles by performance.....	452
Parallel coordinate plot.....	454
Creating a parallel coordinate plot.....	455
Displaying actual values in the parallel coordinate plot.....	457
Summary table.....	458
Box plot.....	459
Comparison circles.....	460
The Visualization types flyout.....	464
Pinning visualization mods to the visualization flyout.....	465
Saving a visualization mod to the library.....	466
Opening a visualization mod that is saved locally.....	467
Duplicating a visualization.....	468
Creating a trellised visualization.....	468
Visualizations trellised by data columns.....	470
Trellis options.....	473
Visualization properties (preview).....	475

<b>Working with visualizations.....</b>	<b>479</b>
Selecting columns on an axis.....	479
Selecting columns using the Data in analysis flyout.....	479
Selecting columns using the column selector.....	482
Other options than columns in column selectors.....	483
Changing the column order on an axis.....	492
Drag-and-Drop.....	492
Column from marked.....	496
Hierarchies.....	503
Defining hierarchies.....	505
Additional operations with time hierarchies.....	510
Editing a defined hierarchy.....	513
Aggregating data.....	514
Adjusting scales on axes.....	517
Adding a zoom slider.....	521
Changing display name for an axis.....	523
Showing/hiding the legend.....	523
Specifying the legend appearance.....	525
Showing gridlines.....	526
Marking.....	528
Marking items.....	529
Marking items using the legend.....	531
Unmarking items.....	532
Reapplying markings when reloading data.....	532
Exploring data across visualizations.....	536
Adding a new marking.....	538
Working with colors.....	539
Changing color of visualization items.....	543
Adding color rules.....	543
Adding color schemes.....	547
Error bars.....	548
Adding error bars.....	549
Information and warnings.....	554
Adding data limitations for a visualization.....	555
Using markings for data limiting.....	558
Limiting the data in a visualization using filterings.....	561
Limiting the data in a visualization using an expression.....	563
Adding a streaming time range to a visualization.....	564
Renderer settings.....	566

Updating the whitelist for allowed URIs when running Spotfire Analyst without a server.....	567
Updating a visualization that needs a manual refresh.....	568
<b>Filtering data.....</b>	<b>570</b>
Filtering data using the Data in analysis flyout.....	570
Filtering data using the Filters panel.....	574
Filter types.....	576
Using range filters.....	576
Using item filters.....	577
Using check box filters.....	578
Using radio button filters.....	579
Using text filters.....	579
Using list box filters.....	580
Using hierarchy filters.....	581
Changing filter type.....	586
Creating a filter transformation.....	586
Filtering schemes.....	587
Adding a new filtering scheme.....	588
Changing the filtering scheme to use on a page.....	589
Filtering in related data tables.....	590
Changing the settings for filtering in related data tables.....	592
<b>Analyzing data.....</b>	<b>596</b>
Displaying item details.....	596
Drilling down into details.....	597
Specifying what to show when no data is marked.....	599
Zooming into visualization details.....	601
<b>Pages and layout.....</b>	<b>603</b>
Arranging visualizations.....	603
Changing the page layout using drag and drop.....	603
Changing the page layout using arrange visualizations.....	604
Arranging pages.....	605
Adding a new page.....	605
Changing page navigation method in an analysis.....	605
Changing the order of the pages in the analysis.....	607
Hiding pages.....	608
Specifying placement of page navigation and status bar.....	609
Viewing analysis pages simultaneously.....	609
Customizing the toolbar.....	610

Changing the aesthetics.....	610
Custom themes overview.....	612
Custom theme settings.....	613
Creating a clean visual appearance.....	616
Adapting layout and panels.....	618
Adapting layout to different screen sizes.....	619
Maximizing the canvas.....	620
Modifying responsive layouts.....	621
Specifying minimum acceptable page size.....	622
Locking the size of certain visualizations.....	623
Authoring analyses for small screens.....	625
Panels and popovers.....	626
Web page panel.....	627
<b>Extending Spotfire.....</b>	<b>630</b>
Developing mods.....	630
Visualization mods.....	631
Action mods.....	633
The Mods development tool.....	634
Actions.....	636
Adding actions to a text area.....	636
Adding actions performed when clicking on visualization items.....	646
Adding an action trigger to a visualization.....	649
Adding a new external action to an analysis.....	650
Configuring action parameters.....	654
Running an external action.....	656
The Actions flyout.....	657
Data functions and the f(x) flyout.....	662
What are data functions?.....	662
Running data functions in Spotfire web clients.....	664
Data functions in the Data canvas.....	668
Interactive tuning of data functions.....	672
The f(x) flyout.....	675
Usage of scripts and data functions.....	678
Manage trust.....	680
Review mod.....	683
Review external action.....	684
<b>Annotations and conversations.....</b>	<b>686</b>
Adding annotations.....	687

Minimizing annotations.....	688
Hiding annotations.....	688
Removing annotations.....	689
Duplicating an annotation.....	689
Collaborating using conversations.....	689
Starting a conversation.....	692
Replying in a conversation.....	693
Deleting a conversation.....	695
Deleting a reply.....	695
Viewing conversations.....	696
Minimizing conversations.....	697
Getting link for sharing.....	697
<b>Using bookmarks.....</b>	<b>698</b>
Adding a bookmark.....	700
Bookmarks example scenarios.....	700
Applying a bookmark.....	705
Updating a bookmark.....	706
Renaming a bookmark.....	706
Deleting a bookmark.....	706
Making a bookmark public.....	707
Making a bookmark private.....	707
Copying a link to a bookmark.....	707
<b>Grouping data.....</b>	<b>709</b>
Tags.....	709
Creating tags.....	711
Removing tags.....	712
Using tags - example.....	713
Creating intervals of numerical data.....	718
Grouping categories.....	720
Grouping from marked categories.....	722
Ungrouping categories.....	726
Handling differently spelled values.....	726
<b>Making your own calculations.....</b>	<b>728</b>
Calculated columns.....	730
Adding a calculated column.....	730
Evaluations of expressions.....	733
Examples of calculated columns.....	735

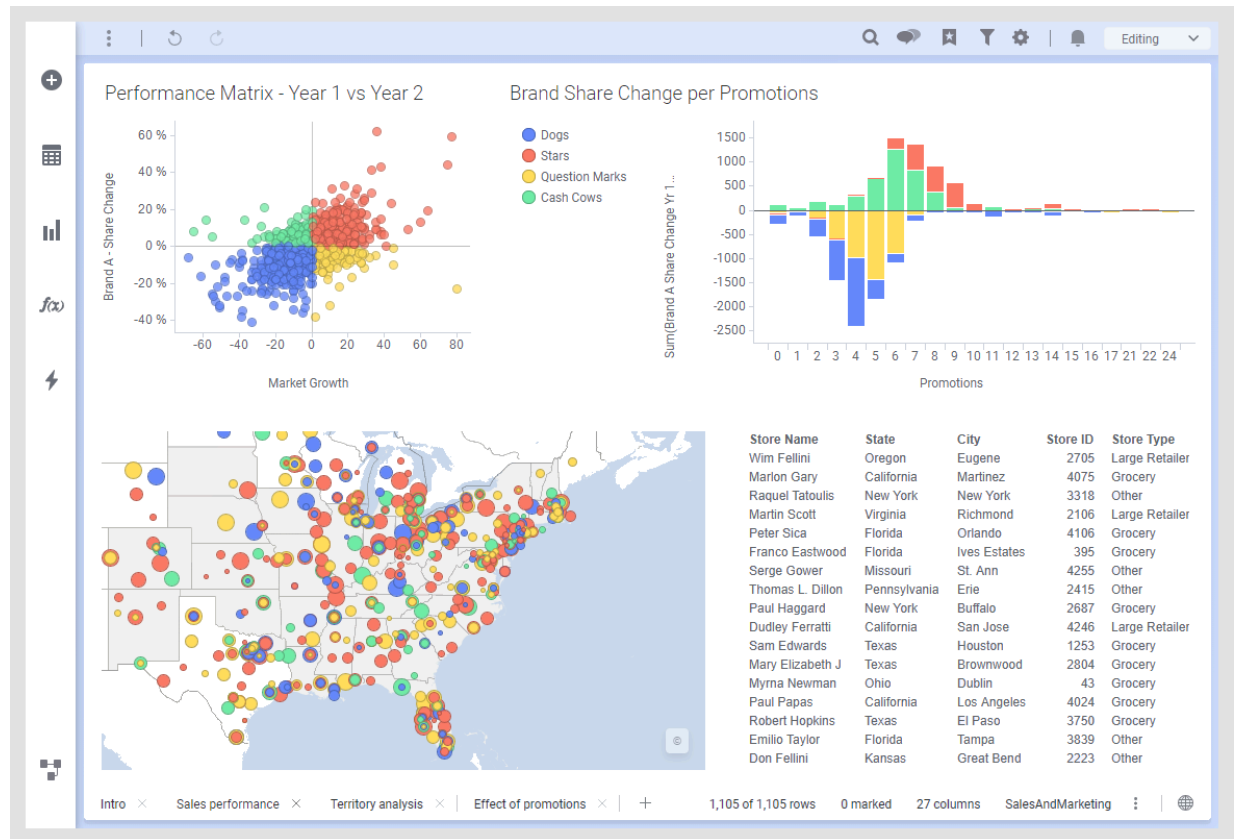
Custom expressions.....	743
Applying a custom expression.....	745
Examples of custom expressions.....	746
Slicing and marker concepts.....	757
Using OVER statement to reference data slices.....	760
Node navigation.....	779
Shortcuts to frequent custom expressions.....	808
Expression language details.....	846
Operators.....	849
Functions.....	852
Data types.....	904
Properties in expressions.....	907
Aggregations and statistical measures.....	920
Sum and Product.....	921
Average.....	922
Count.....	922
Unique Count.....	922
Min and Max.....	923
Median.....	923
Standard Deviation.....	923
Standard Error.....	924
Variance.....	924
Confidence Intervals.....	924
Percentiles and Quartiles.....	925
Adjacent Values and Outliers.....	927
Concatenate and Unique Concatenate.....	928
First and Last.....	928
Geometric Mean.....	928
Mean Deviation.....	929
Median Absolute Deviation.....	929
Most Common.....	929
<b>Exporting your analysis.....</b>	<b>930</b>
Exporting to PDF.....	930
What to export.....	931
General export settings.....	932
Exporting an entire analysis.....	934
Exporting a visualization.....	934
Exporting a visualization and include non-visible content.....	935
Exporting a single analysis page.....	936

Exporting specific analysis pages.....	936
Exporting Details-on-demand.....	937
Exporting a prepared report.....	938
Prepared reports.....	938
Exporting to Microsoft PowerPoint.....	949
Exporting to Microsoft Excel.....	950
Exporting visualization to image.....	951
Exporting data.....	951
Exporting data to the library.....	951
Exporting data from a visualization to file.....	952
Exporting data from the data table to file.....	953
<b>Find.....</b>	<b>955</b>
Searching in Spotfire clients.....	956
<b>Saving an analysis.....</b>	<b>959</b>
Deleting items from the library.....	960
Reverting to a previous version of a file.....	961
Adding library items as favorites.....	962
Creating analyses for others.....	962
Dashboards.....	962
Designing accessible dashboards.....	971
Reports.....	972
Guided analyses.....	973
Preparing analyses for Spotfire web clients.....	975
Adding functionality for mobile devices.....	977
<b>Sharing analyses.....</b>	<b>987</b>
Copying a link to an analysis.....	987
Copy link dialog.....	988
Links to analyses in the library.....	989
Downloading an analysis.....	992
Optimizing a dashboard using URL parameters.....	993
<b>Downloads.....</b>	<b>994</b>
<b>Spotfire Documentation and Support Services.....</b>	<b>995</b>
<b>Legal and Third-Party Notices.....</b>	<b>996</b>
<b>Index.....</b>	<b>1000</b>



# This is Spotfire

Do you want to do ad-hoc analytics, perform in-depth exploration of data, or create informative, up-to-date dashboards? Then Spotfire® is the tool. In this guide, you learn what Spotfire is, and how to get the most out of it. You will find what you can do to gain insights about your data, and how to do it in detail, all supplemented with examples that serve as a source of inspiration.



Read on to understand how to get the data into an analysis, and how you can visualize and interact with the data to answer the questions you might have. Interactions can involve anything from filtering to data of interest, drilling down to more detailed levels, performing calculations on the data, or applying statistical methods.

The Spotfire user interface assists you in analyzing the data by providing recommendations on visualizations that fit the data you want to display. You can also get assistance in discovering relationships in the data, or how to add new data in good ways. Recommendations and tips are indicated using a light bulb icon.

Your findings can be shared with others in different ways, and you have a lot of options to customize the visual appearance of an analysis.

In addition, you learn how to know the details of your data, and how to keep track of the data history from the initial loading to the current content after modifying and cleansing it.

## Get started

[The user interface](#)

[Load data](#)

[Visualize data](#)

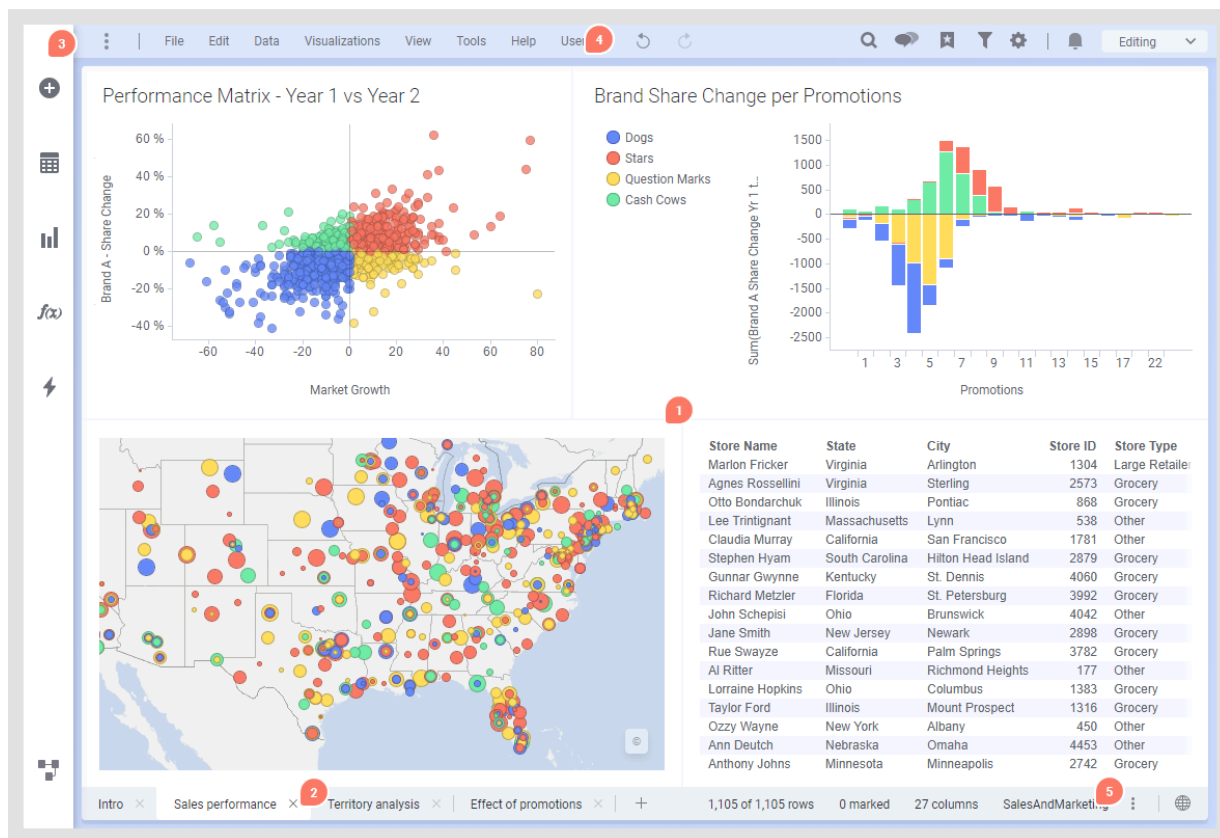
Analyze data

Make calculations on the data

Share with others

## The user interface

The most vital parts of the user interface are shown below.



### 1. Visualizations

An analysis consists of visualizations based on the loaded data. To provide the best representation of your data, many visualization types are available. Within each visualization type, you can use various properties to reflect different dimensions of the data, for example, color, size, and shape.







### 2. Pages

The visualizations are organized onto pages in the analysis. You can resize and move around the visualizations on the pages to get the desired layout when creating an analysis.

A page can also contain text areas (added using the installed client). There you can add information, static as well as dynamic, and also initiate various kinds of interactions like filtering or switching to another analysis page.

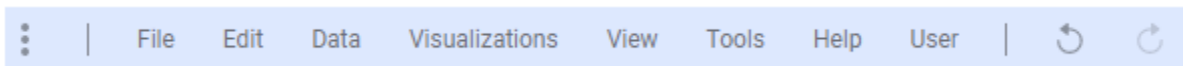
### 3. Authoring bar

If you have authoring permissions, you access the most frequently used features for authors by clicking the icons on the authoring bar. If the authoring bar is unavailable, switch from Viewing mode to Editing mode in the top right corner of the menu bar, see below.


- **Files and data** : Here you add the data you want to visualize, or open existing analyses.
- **Data in analysis** : Lets you create visualizations by selecting the data you are interested in first, and then you let Spotfire recommend various ways to visualize it. This is also where you can get more information about the data and filter out values.
- **Visualization types** : Here you create visualizations by selecting the type of visualization you want to use. Then you configure the visualization your way.
- **f(x) – analytic tools** : If enabled, gives you easy access to your favorite data functions, which can perform different calculations. In the installed client, you can also reach analytic tools from here.
- **Actions** : Here you get an overview of the action mods and external actions in your analysis. You can run an action from the flyout, and edit its configuration.
- **Data canvas** : Here you get a view of the data structure in the analysis, and you can also make changes to it. You can, for example, add more rows and columns to data tables, transform the data (installed client only), and specify when and how the data should be loaded and stored.

#### 4. Menu bar

In the left part on the menu bar, features mentioned in the authoring bar section above are accessible, but you can also get access to other important features, for example, export and calculation options, as well as the user guide. In addition to the menus, buttons for Undo and Redo are available.




If you want, you can click the three dots furthest to the left to show or hide this part of the menu.

To the right on the menu bar, you can use Find  as a short way to find contents as well as actions and features you are looking for. This part of the toolbar is also where you are given information, various notifications, and scheduled updates details, and you can open the collaboration, bookmarks, and filters panels, and settings for the visualization properties. You can [customize](#) this part of the toolbar by selecting shortcuts to actions and tools that you use often.



If you are authoring analyses to be consumed by users who do not have authoring rights, you get their view of the analysis by switching from Editing to Viewing in the drop-down list.

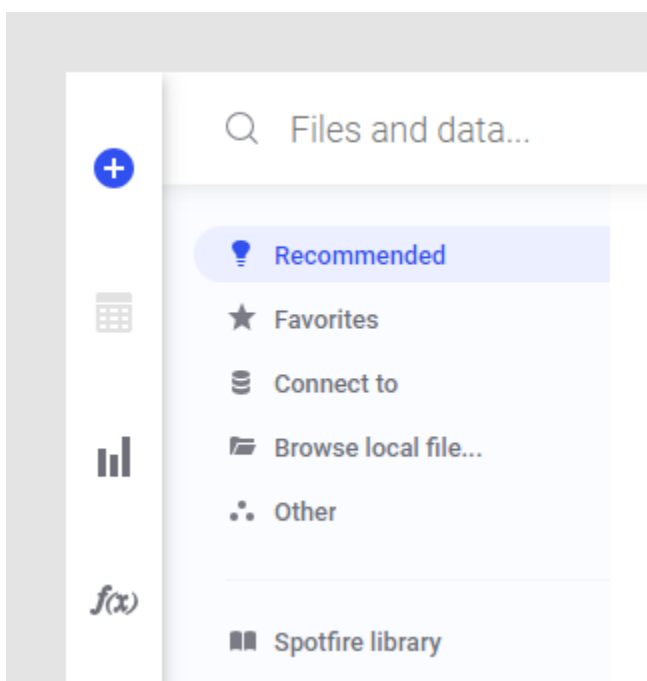
#### Status bar

The status bar is located at the bottom or the top of the window and shows information about the main data table used by the active visualization (the visualization that was last clicked upon). If the information is not visible, click .

## Load data

The most important thing when working with Spotfire is your data. All data is accessed by clicking **Files and data** on the authoring bar.

If you are a consumer user you will only have access to predefined analysis files, whereas author users can access data from multiple different sources. You reach all the different ways to add data via the **Files and data** flyout (reached by clicking + on the authoring bar).



Use the search field at the top of the flyout to find data from the library or in databases. Use the **Browse local file** option to locate files on your computer.



You can also drag a supported file to the analysis directly and drop it in your browser or installed client.

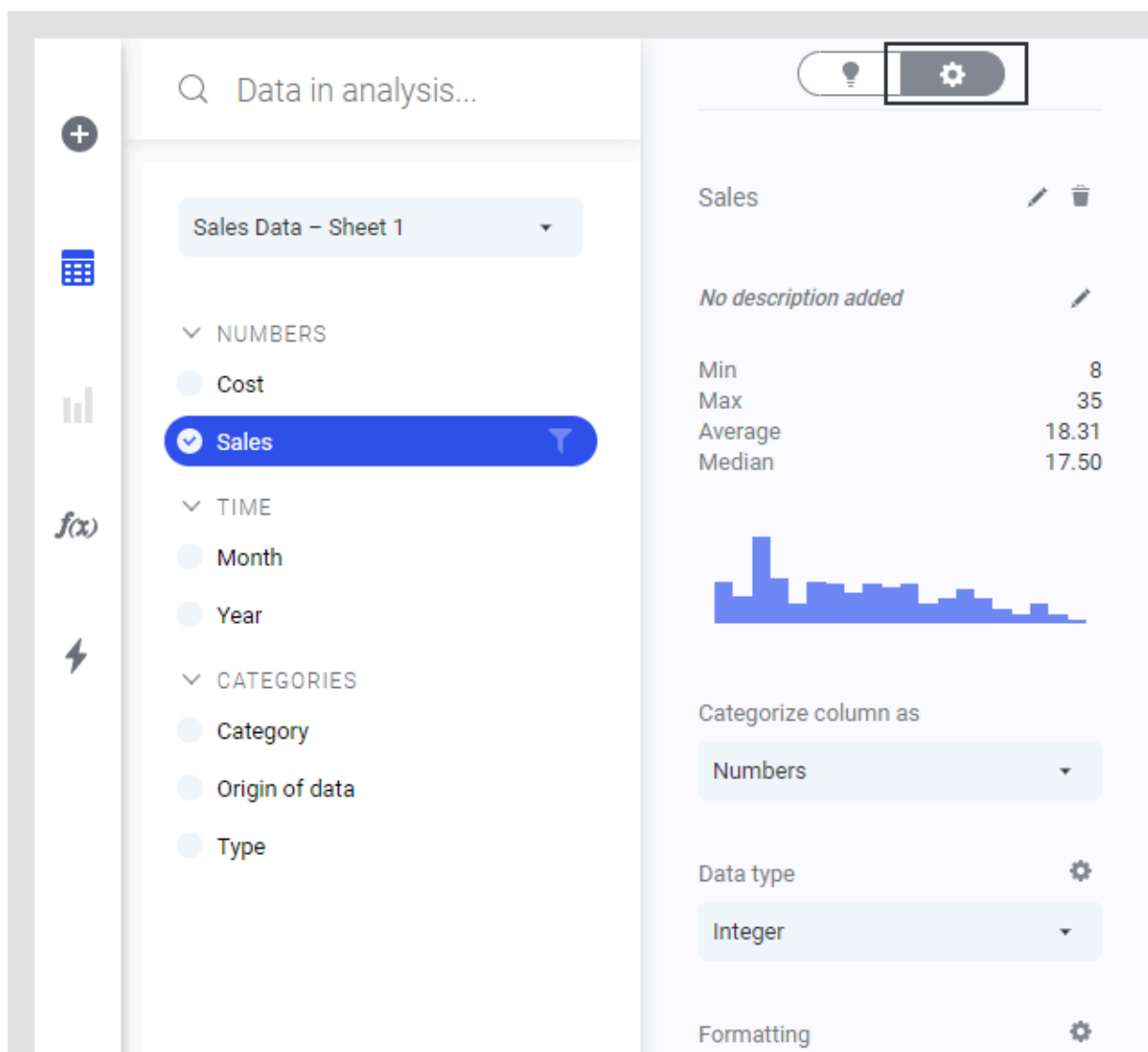
For more information, see [Loading data](#) on page 38.

## Know the data

As an analysis author, you can investigate the metadata and learn more about the columns in your analysis. Learn more by inspecting your data in the **Data in analysis** flyout and in the data canvas.

The **Data in analysis** flyout is automatically opened when you add a new data table. It shows you a structured list of all columns in the data table, where different types of columns are grouped into categories like NUMBERS, TIME, and so on. Numeric data is often used as summarized data on axes in a visualization, whereas categories are used to split the data in different groups. It is possible to change the categorization if the automatic attempt is not good enough. The behavior of a column when added to a visualization differs depending on how it is categorized.

To know the details of a data column, you can select it in the flyout, and click **Details on selected column** as shown below. Now you can see some metadata, such as the Min, Max or Average value in a numeric column, or the unique count of categories. You also get access to some data preparation and cleansing tools, which can help getting visualizations the way you want them. For more information, see [Data in analysis](#).



You can also get more information about the data in the **Data canvas**, which is accessed by clicking the button at the bottom of the authoring bar.

The screenshot shows the Spotfire Web Client interface. At the top is a menu bar with options: File, Edit, Data, Visualizations, View, Tools, Help, User. Below the menu is a toolbar with icons for search, share, filter, settings, and notifications. The main area is a data canvas titled 'Sales Data'. It contains a flow diagram with three nodes: 'Sales Data.xlsx - Sales Data', 'Added rows', and 'Sales Data'. The 'Sales Data' node is highlighted with a blue border and contains the text: 'Sales Data. This is the data table used in the analysis.' Below the flow diagram is a preview of the 'Sales Data' table. The table has 7 columns: Year, Month, Category, Type, Sales, and Cost. The data is as follows:

Year	Month	Category	Type	Sales	Cost
2022	December	Vegetables	Cucumber	12	10
2022	December	Vegetables	Tomatoes	11	10
2022	December	Vegetables	Lettuce	17	21
2020	January	Fruit	Apples	12	10
2020	January	Fruit	Pears	21	13
2020	January	Fruit	Bananas	29	26

The table has 432 rows and 7 columns. The bottom right corner of the canvas shows a globe icon.

Here you can see how each data table in the analysis was created, and you can add data or edit settings for different data sources. Click each node for details about that source or operation. For more information, see [Data canvas](#).

By looking at the **Data** tab in the lower right part of the data canvas, you can see a preview of the imported columns.

INFORMATION	DATA	COLUMN PROPERTIES	DATA TABLE PROPERTIES		
Month	Category	Type	Sales	Cost	
String	String	String	Integer	Integer	
January	Fruit	Apples	12	10	
January	Fruit	Pears	21	13	
January	Fruit	Bananas	29	26	
January	Vegetables	Cucumber	9	6	
January	Vegetables	Tomatoes	13	11	
January	Vegetables	Lettuce	22	20	
February	Fruit	Apples	11	9	
February	Fruit	Pears	21	13	
			226 rows	9 columns	

On the **Column properties** and **Data table properties** tabs, you see the current value of the properties, and for the final data table node, you can [change the value of a property](#).

INFORMATION

DATA

COLUMN PROPERTIES

DATA TABLE PROPERTIES

Q

Type to search

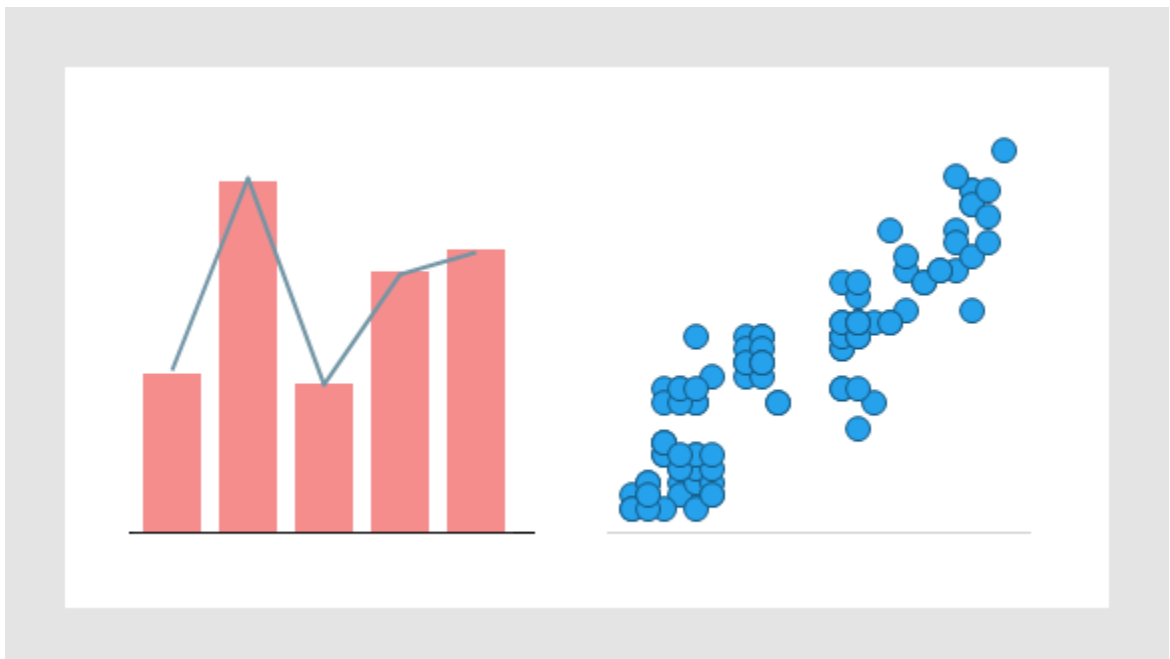
Edit properties

Name	ColumnType	DataType	ExternalNa...	ExternalId	IsValid
Id	Imported	String	Id	Id	True
ParentId	Imported	String	ParentId	ParentId	True
Type	Imported	String	Type	Type	True
CreatedById	Imported	String	CreatedById	CreatedById	True
CreatedDate	Imported	DateTime	CreatedDate	CreatedDate	True
IsDeleted	Imported	Boolean	IsDeleted	IsDeleted	True
LastModifie...	Imported	DateTime	LastModifie...	LastModifie...	True

## Visualize data

To represent the data visually, many different types of visualizations are provided, all with different strengths for displaying the data in an informative way.

For example, there are visualizations suitable for making comparisons and investigating distributions, correlations or trends in the data as exemplified below. Whether the data is continuous or categorical, if it should be aggregated, and whether one, two, or more variables need to be represented, are also of importance when deciding which visualization to use.



When you add a visualization to an analysis, you can do it in three ways:

- **Start from the data**

By selecting data columns you are interested in among the loaded data, you will get recommendations on visualizations based on these particular columns. Then, the visualization of interest can simply be dragged to the analysis page, and if needed, you can fine-tune it there.

When using the installed client, beyond basic visualizations based on selections, you might also get visualizations recommended to you based on insights calculated by the recommendations engine. Such an insight could be that another, not selected data column, has a strong correlation to any of your selected columns.

- **Start from a visualization type**

You can simply select the visualization type you want to use, and drag it to the analysis page. Once on the page, you can select which data columns you want to visualize, and how. This way you build the visualization from scratch.

- **Search for what you want to visualize**


You can search for data you are interested in among existing content and also get recommendations on new visualizations that can be relevant. The visualizations can then be dragged to the analysis page.

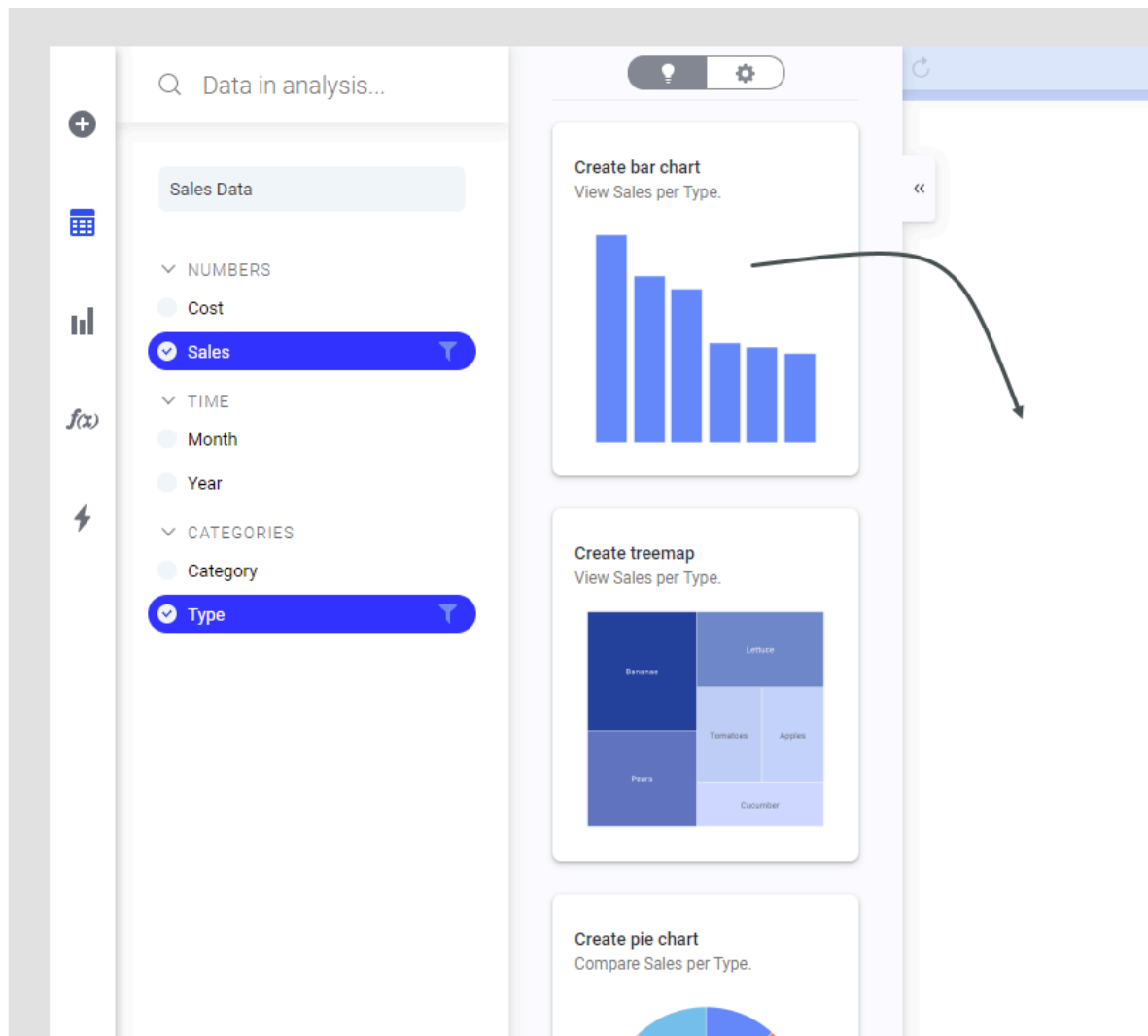
No matter method used, you can at anytime modify the properties of the visualization. The properties to change can be anything from changing what data columns are visualized, changing the aggregation method used to summarize the data, to changing colors and fonts. You can even switch to another visualization type.


## **Get started**

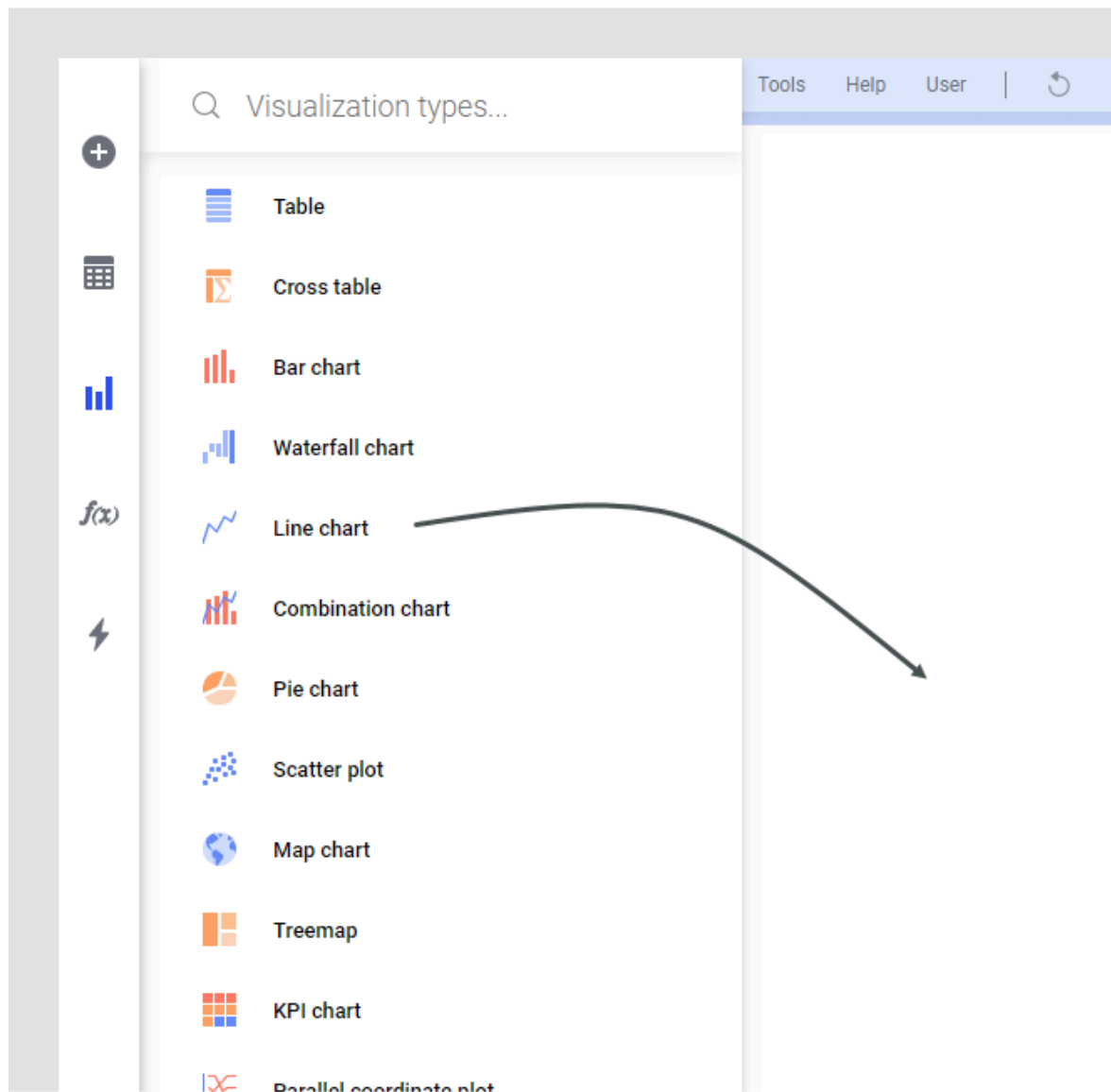
To add a visualization, do one of the following:




- Click **Data in analysis**  on the authoring bar, and in the opened flyout, select the data columns you are interested in. Among the visualizations that are recommended, click or drag the one you are interested in to the analysis, and explore it.

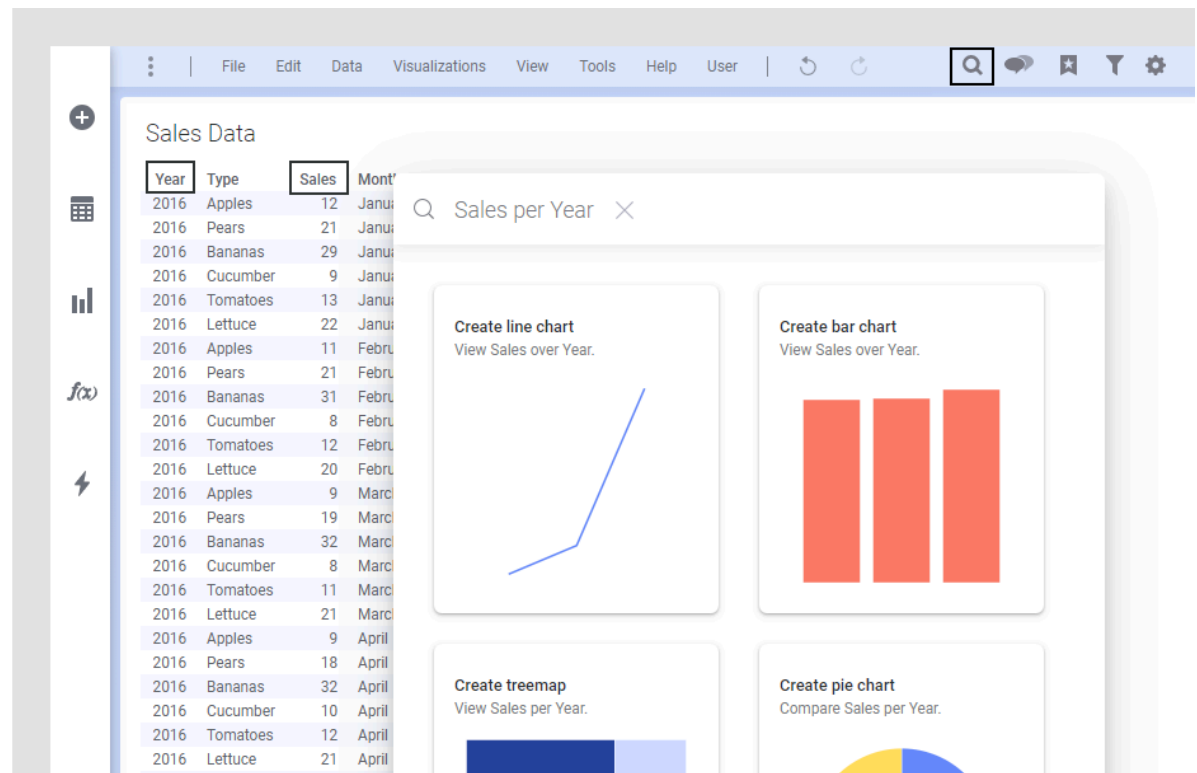


- Click **Visualization types**  on the authoring bar, and from the opened flyout, click or drag the visualization type you want to use to the analysis page. Then modify the default settings to make the visualization look the way you like.



- Click **Find**  on the menu bar. In the opened search field, type what you want to visualize, and let us suggest visualizations for the matches found. For example, if you type the names of data columns

you want to explore, visualizations are suggested as shown in the image below. Then click or drag the one you are interested in to the analysis, and explore it.



For more information on what you can search for, see [Find](#).

## Analyze data

Often, the visualizations you add using the Data in analysis flyout provide enough knowledge of your data, and no further exploration is needed. But, after you have added the visualizations to your analysis, there is a wide range of possibilities to extend the analytics, no matter which method you used for creating them.

### Summarize data

A central part of the analysis is to decide how the data should be represented by the visualization items, that is, which parts the data should be split into, and which summarized values the items should represent. For the summarization, you have a rich selection of methods to choose from, where the most common aggregation methods are sum and average.

### Interact with the data

Not only do separate visualizations let you analyze your data; the visualizations are also by default connected to each other, so your interaction with one visualization will spread to other visualizations. This way you can compare different views of the same data.

There are different ways to interact with the data, and by doing so, you get more knowledge of the data:

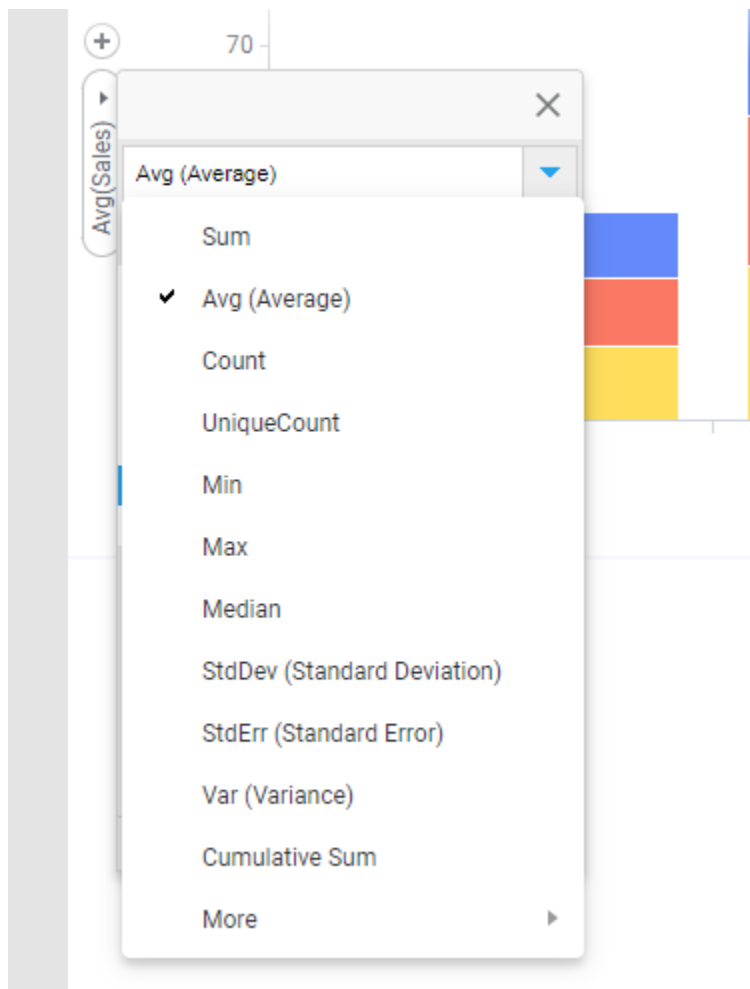
- You can **filter** to data values of particular interest, and base the visualizations on these values only.

- You can get more details of the data in various ways:
  - Hover with the mouse over visualization items to see details in tooltips.
  - Mark items in one visualization, and items associated with the ones you marked will get marked in other visualizations as well.
  - Use [details visualizations](#), that is, mark items of interest in a visualization to form a subset of the data, and create a visualization that represents this subset only. This can be iterated in the new visualization, which enables you to drill down into subsets within subsets of your data.
- You have different methods to **create your own groups of data**. This is convenient when you want to analyze the data differently from using the ordinary way of grouping data by selecting columns on the different axes such as the X-axis, or the color axis. Examples of methods are [binning data](#) and creating [subsets](#) of data. In the installed client, you can also assign tags to data or create lists of data you want to track.
- You have several options to [make calculations](#) on the data to retrieve more information.


## Get started

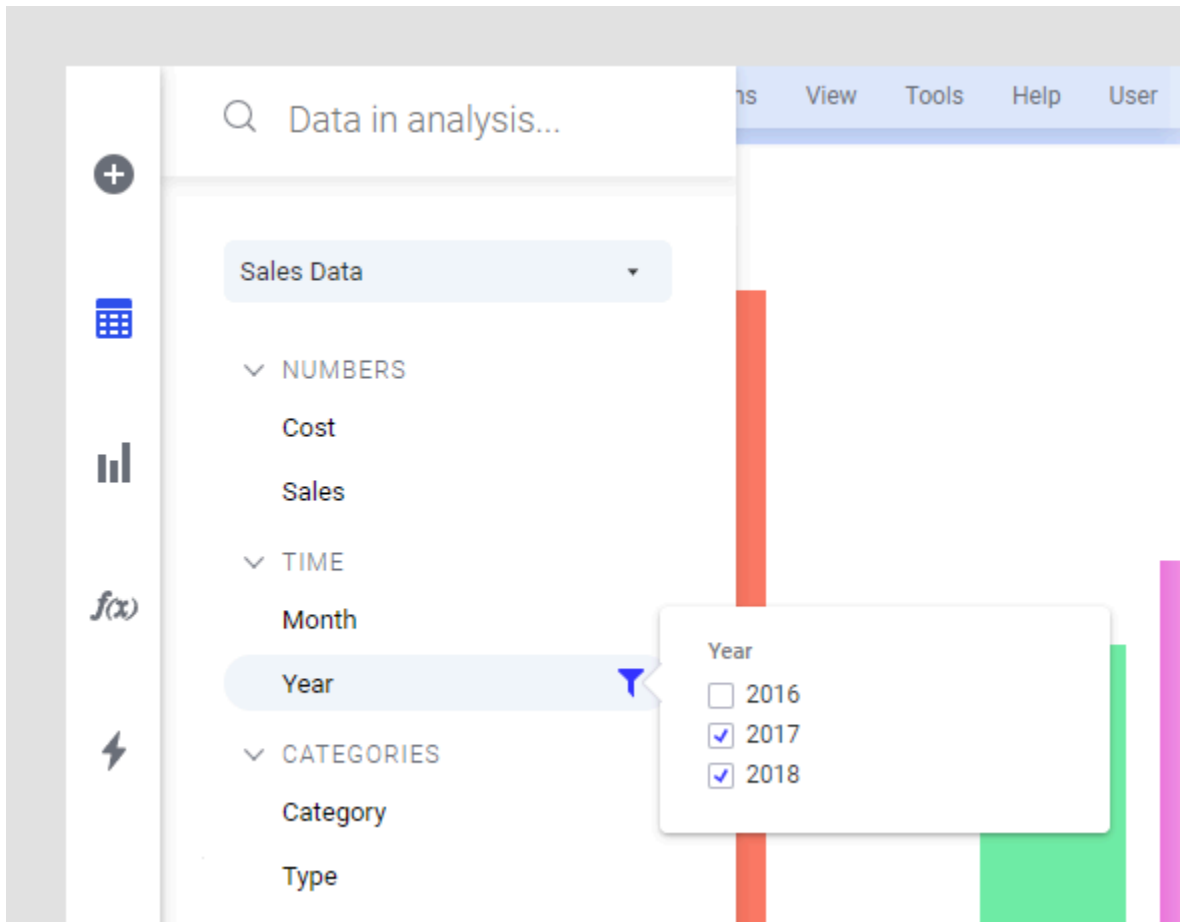
### Get started with summarizing data

To specify how the data in the column you have selected should be summarized, select an aggregation method on the column selector. In this example, the Sales values are summarized into an average.



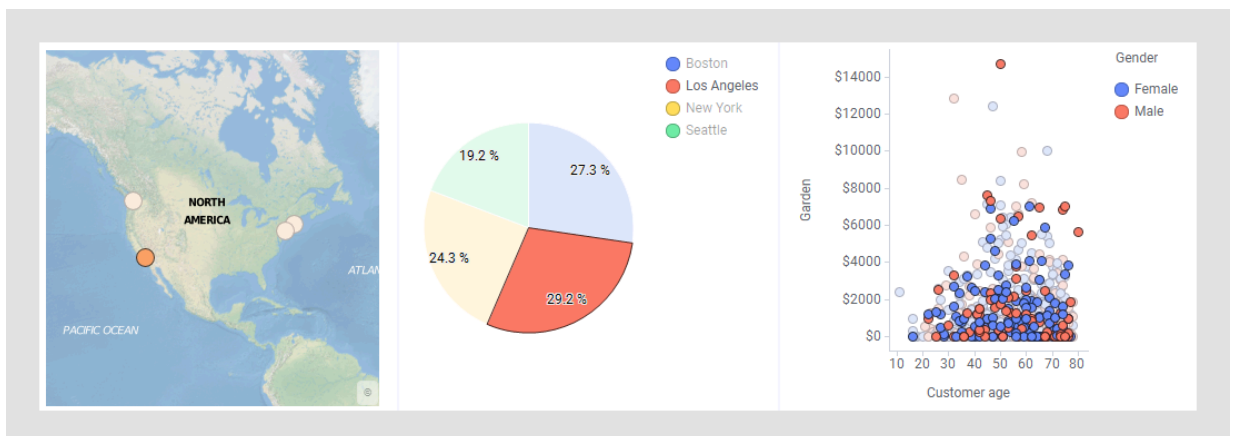
### Get started with filtering

To filter to certain values in a data column, you hover with the mouse pointer over the column in the **Data in analysis** flyout, and click **Show filter** . Then you adjust the filter.



### Get started with viewing details

To drill-down into details, you mark items of certain interest in one visualization, and corresponding data in the other visualizations based on the same data table will by default be marked automatically. For example, if 'Los Angeles' is marked in the map chart below, 'Los Angeles' items get marked also in other visualizations. In the pie chart, the 'Los Angeles' sector gets marked, and in the scatter plot, 'Los Angeles' customers.



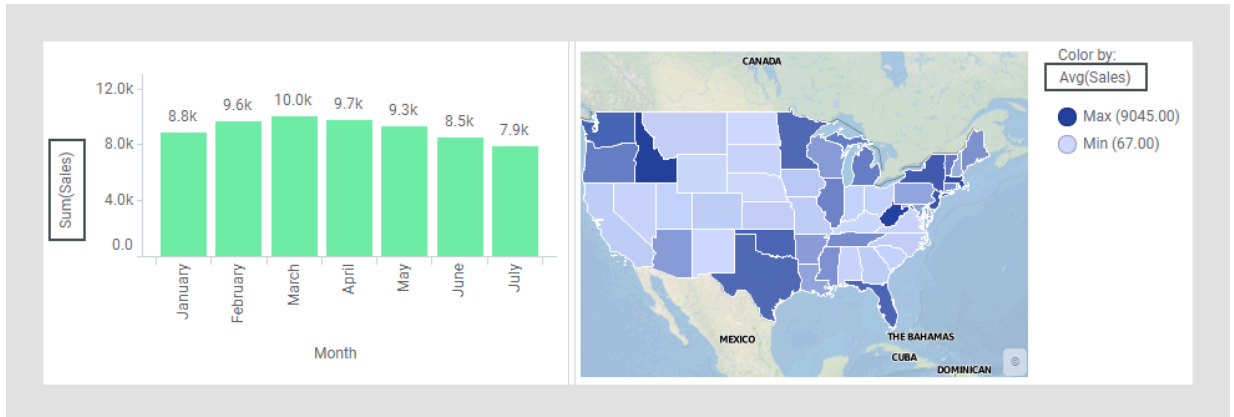
To drill down into subsets within subsets of your data, see [Drilling down into details](#) on page 597.

## Make calculations on the data

There are many ways to make calculations on the data.

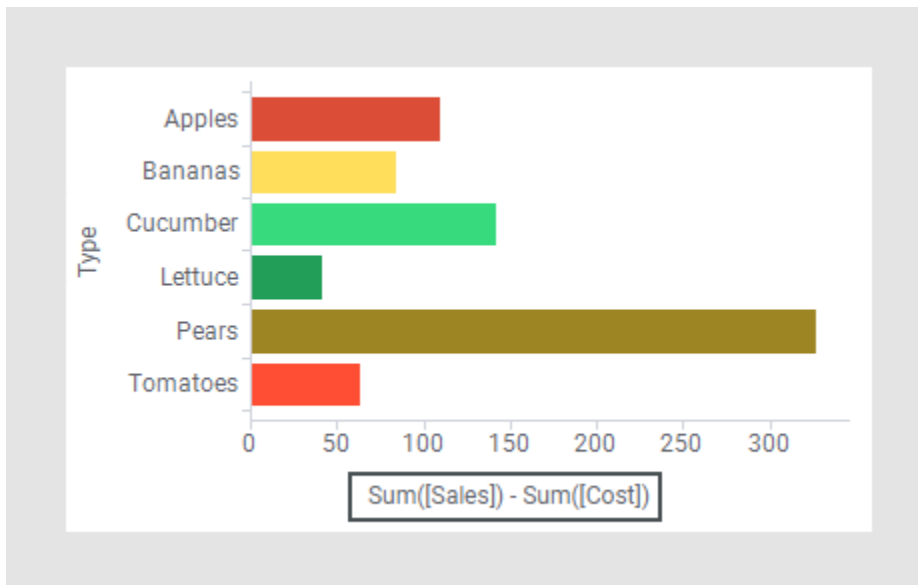
### Aggregations and expressions

The most typical calculations that are made in the visualizations are the [aggregations](#) (for example, sum, average, max, and median), which you specify on the various visualization axes to summarize the data:



However, calculations on the data can be done in other ways and at different places in an analysis to get more out of your data.

First of all, if any of the predefined aggregations does not match the calculation you want to do, you can create your own [expressions](#) to apply on the various axes in a visualization as exemplified below.



Moreover, you can create expressions whose resulting values are added as a [new column](#) in a data table.

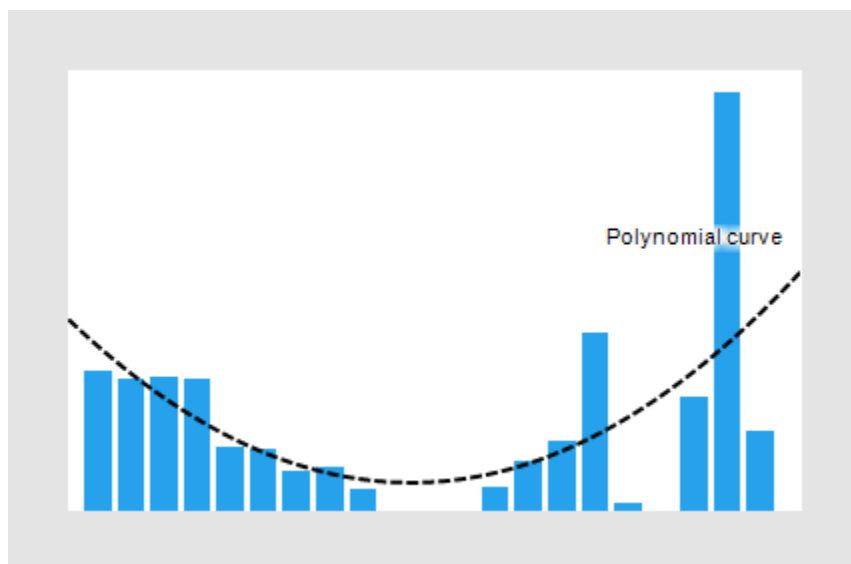
Expressions, predefined as well as new expressions, can be applied not only on the ordinary axes that are used for setting up a visualization. You can also use them, for example, to add [error bars](#) or to create [subsets](#).

## Statistical tools

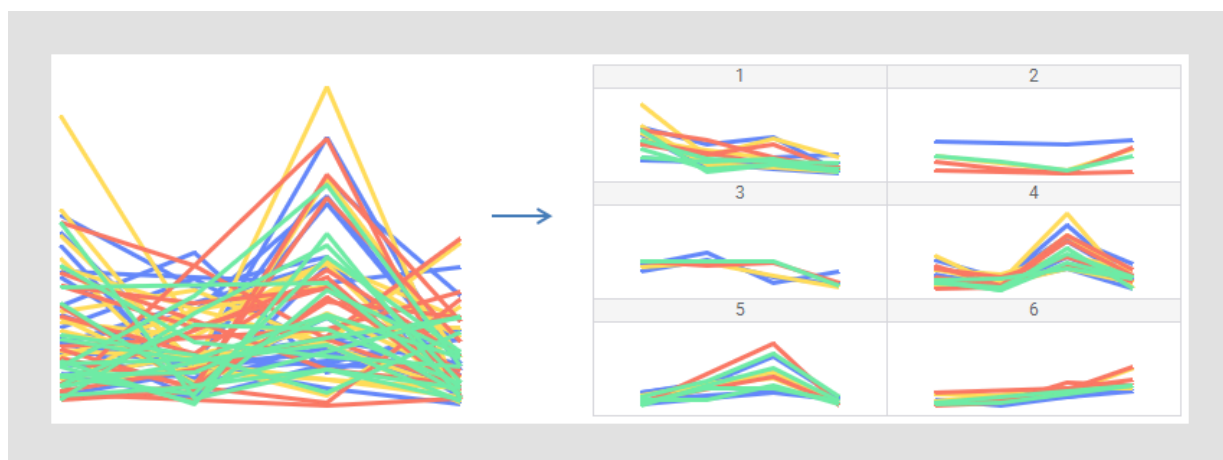
When using the installed client to create analyses, there are many built-in tools for performing powerful computational and statistic analytics. Curve fit models, methods for examining data relationships, data clustering methods, ranking of line similarities, or making predictions are all built-in tools.

The images below illustrate a couple of examples.

### Curve fit



### K-means clustering



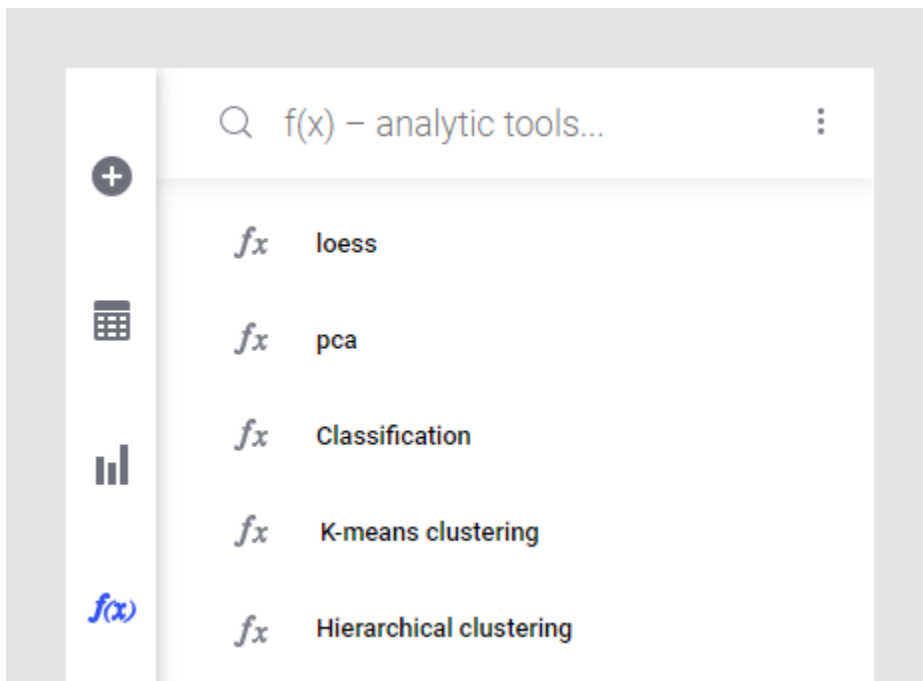
## Data functions and scripts

The use of [data functions](#) is a way of adding even more statistical capabilities to an analysis. Data functions are calculations based on scripts running under Spotfire® Enterprise Runtime for R (a/k/a TERR™), open source R, Python, or other solutions. Which of these options are available depend on your specific Spotfire environment. The data functions are defined using the installed client but they can be used and run from within various parts of Spotfire in all clients.

Using the installed client, you can also define IronPython scripts, JavaScripts, or scripts written in custom query languages, to perform actions in your analysis. The scripts have access to the Spotfire API.

## The f(x) flyout

The [f\(x\) flyout](#) is a place where it is possible to pin data functions saved in the library, for easy access to calculations and tools when creating analyses. In the installed client, you can also find analytic tools such as K-means clustering here.



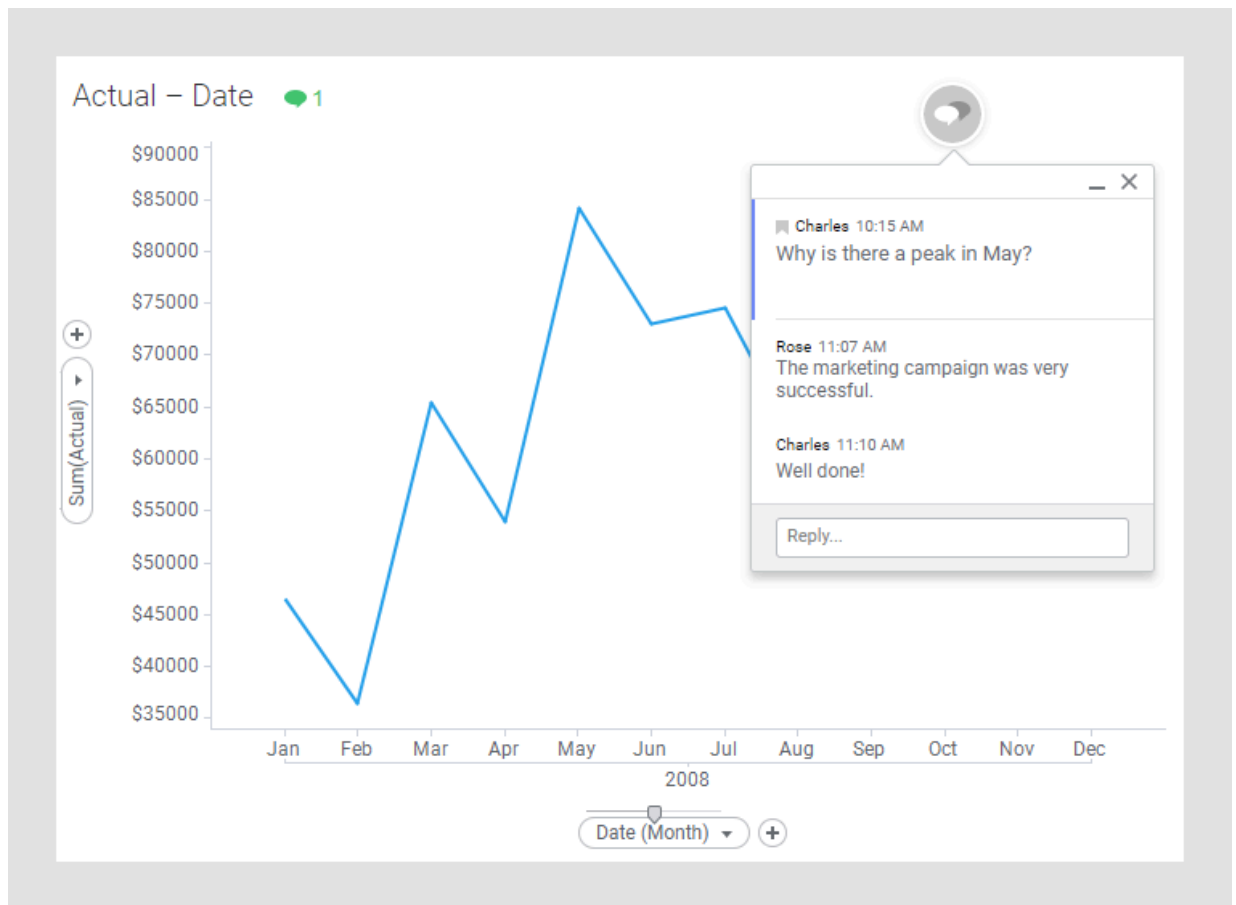
## Share with others

Analyses can serve different purposes. Some might be used for performing advanced analytics, and some as display of informative summaries in dashboards, some are intended for a small group of people and some for a big audience. To serve different users' needs, for one-way as well as two-way communication, there are a number of possibilities available.

- You can [save an analysis to the library](#). Then you and your colleagues can work together on it, keeping everyone up to date.
- While working on an analysis, you and your colleagues can have a conversation by adding [comments](#) into the visualizations, or, if it is a one-way communication, you can add [annotations](#), for example, to guide others through the analysis.
- You can capture [bookmarks](#) of analysis views with interesting findings. Other people, or you, can then easily return to these specific views.
- You can [export](#) an analysis to PowerPoint and to a PDF document. An ad-hoc PDF export results in a nice output, but you can also, by using more advanced export settings, prepare PDF reports.
- By applying customized [visual themes](#), you can create good-looking analyses to be consumed, for example, on dashboards.
- The responsiveness of the analysis layout makes the experience nice also when consuming analyses on [smaller screens](#).

The image below exemplifies a conversation within an analysis:





## Spotfire clients and user roles

When you are working with Spotfire as an author, you can create analyses either using the installed client or using the web client. Even though much of the functionality is available in the web client, the installed client allows you to do more.

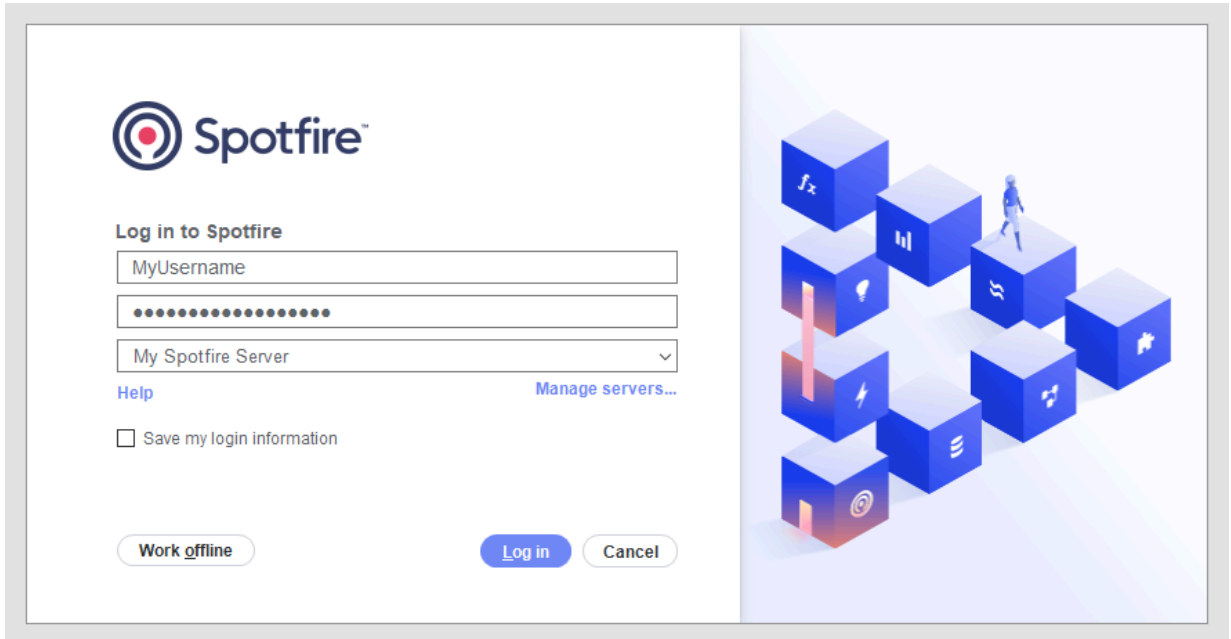
However, which functionality you have access to in different environments can also depend on which license features you have been assigned by your administrators.



This user guide contains information about all functionality that can be used in the web client. It also includes some of the functionality that is only available in the installed client, to make it clearer how someone with access to the installed client would author or configure that functionality. If you do not have access to all licenses, some tools and options described in this help will be unavailable. For a list of the available licenses and features, see the [License feature reference](#) in the *Spotfire® Server and Environment - Installation and Administration Manual*.

## Logging in to the installed Spotfire client

When you start the installed Spotfire client, a login dialog is shown. Enter your username and password, and then click **Log in** to start Spotfire. If it is the first time you start Spotfire, follow the instructions in the dialog to complete the installation of the application.



If you select the **Save my login information** check box in the login dialog, you will automatically be logged in the next time you start Spotfire. If you want to reach the login dialog again when starting Spotfire later on, you can force it to be shown by searching for **Spotfire** from the Windows start menu, and then selecting the **Spotfire Analyst (show login dialog)** option. Logging in to Spotfire will give you access to the joint library and other collaboration features.

### Managing servers

If you are working for a larger company with multiple Spotfire Servers, you may occasionally also need to change the server you are connecting to via the drop-down list. New servers can be added to the list by clicking on the [Manage servers](#).





If you are connecting via proxy server, you may need to change your security settings in your web browser prior to logging in to Spotfire. See the help of your browser for more information. Before logging in to Spotfire, make sure that the Spotfire Server start page can be accessed by browsing to `http://<hostname>/spotfire/`.

### Making sure the connection to the server is secure

To ensure that the communication is encrypted when you connect to a Spotfire Server, you must connect to the server over HTTPS. If you connect to a server without using HTTPS, sensitive information, such as your username and password, is not encrypted.

To be able to connect securely, the Spotfire Server must be configured to enable HTTPS for communication with the Spotfire clients. For more information about configuring the server, see [Configuring HTTPS](#) in the Spotfire Server Installation and Administration guide.

If you add a server address that does not use HTTPS, the server name is highlighted with a notification icon  in the login dialog. This is to make you aware that the connection to that server will not be secure.

## Deployment areas

If your Spotfire administrator has configured a server with multiple different deployment areas, and your user identity allows you to use more than one of these areas, you will be asked to select deployment area when logging in. To learn more, see [Select deployment area](#).

## Downloading updates

Spotfire automatically checks for updates that apply to you on your Spotfire Server. If you have a network connection to the Spotfire Server, and there are updates available, you will be **notified** when you are logging in. You can get a look at the contents of the available updates by clicking on the **View updates** link, and then decide whether to install the updates or not.

## Working offline

If you are not connected to the network where your Spotfire Server is located, you may still be able to work with Spotfire offline, depending on your company's settings. Almost all of the functionality of Spotfire works fine without a connection to the server. However, you will not have access to the library, which means that analyses and data stored there, as well as information links or shared data connections to databases, will be unavailable. To work offline, simply click the **Work offline** button in the login dialog. With some licenses of Spotfire, you do need to connect to your Spotfire Server at least once a month to be able to continue to work offline.

## Updates and working offline

If you have more than one server, and for one of them, the login has provided you with updates, the server must be selected in the login screen for those updates to be available, even if you choose to work offline.

## Running Spotfire without a server

If you do not have access to a server at all, you can run Spotfire without a server. The same functionality will be available as when working offline, but to gain access to any updates in Spotfire, you must manually download and install the newer version of Spotfire. To run without a server, select **No server**

in the login dialog the first time you start Spotfire, and then click **Continue without server**. The next time you start Spotfire, you will not see a login dialog. If you want to add a server later on, you can force the login dialog to be shown by searching for **Spotfire** from the Windows start menu, then selecting the **Spotfire Analyst (show login dialog)** option, and clicking **Manage servers** in the login dialog.



When running the installed client without a server (installed using the `setup-single-user-@product.version@.exe` installer), to be able to use geocoding, you must also install the default geocoding tables using a separate installer. Also, to run the client on different languages, you must run the separate installer for the desired language, as described in the readme file available with your installation kit.

## Command line parameters

It is possible to configure the launch behavior of Spotfire Analyst using command line parameters. See the command line parameter reference on the [Community](#) for a list of all available command line parameters.

## Manage servers

The Manage servers dialog is shown when you click on **Manage servers** before logging in to the installed Spotfire client.



The following terms in the dialog are explained below:

Option	Description
Available Spotfire servers	Lists all the previously added Spotfire servers, which you can select to log in to.
Add	Opens the <a href="#">Add server view</a> of the login dialog, where you can add new Spotfire servers to the list.

Option	Description
Edit	Opens the <a href="#">Edit server view</a> of the login dialog. Here, you can edit the display name and address of a server, and view the deployment area of the selected Spotfire server, provided that your administrator has granted you access to multiple deployment areas (and you have selected to remember the area).
Delete	Deletes the selected server from the list of available Spotfire servers.
Server details	Lists information about the selected Spotfire server, such as its address, the preselected deployment area (if one is <a href="#">remembered</a> ), authentication type, and version number.

## Add/Edit Server

In the Add/Edit Server dialog, you can specify the web address of the new server, type a custom display name, or, depending on the groups assigned to you by your Spotfire administrator, you might be able to clear a preselected deployment area on the server.

### Prerequisites

Add/edit server can only be done in the installed client.

### Procedure

1. To reach the add server view, click the **Manage servers** link in the login dialog of Spotfire.

2. In the Manage servers view, click **Add**.

To reach the Edit server view, click the **Manage servers** link in the Spotfire login dialog, select a server and click **Edit**.

3. Change the settings in the dialog the way you want it.

Option	Description
<b>Spotfire server address</b>	This is where you specify the web address to the new server. You might need to specify a port number. Contact your Spotfire administrator for this type of information.
<b>Server display name</b>	If desired, you can type a custom display name to show in the drop-down list of the login dialog, instead of showing the actual server address. This can be useful in environments with multiple servers and long server addresses.
<b>Preselected deployment area</b>	This option is only available if you have selected to remember a deployment area.


Option	Description
	<p>Depending on which groups your Spotfire administrator have assigned you to, you might have access to multiple deployment areas on the server.</p> <p>Deployment areas are configured by the Spotfire administrator and make it possible for your company to give different users access to different versions of the Spotfire client, while still using a single Spotfire server. For example, this can be important if a new version of the client has been released and the administrator wants a limited number of users to try the new version before it is taken into production.</p> <p>If you have selected to remember the deployment area used at your last login, then the name of the deployment area is shown here.</p> <p>If you need to change to a different deployment area, click on the link at the bottom of the dialog to clear the preselected deployment area. The <a href="#">Select deployment area dialog</a> will then be shown again upon next login, provided that your group access still allows multiple deployment areas.</p>

## Select deployment area

The Select deployment area dialog is shown if your Spotfire administrator has configured a server with multiple different deployment areas (for example, to allow a limited number of users test a new Spotfire version while others still use the old version) and your user identity has been granted access to more than one of these areas.

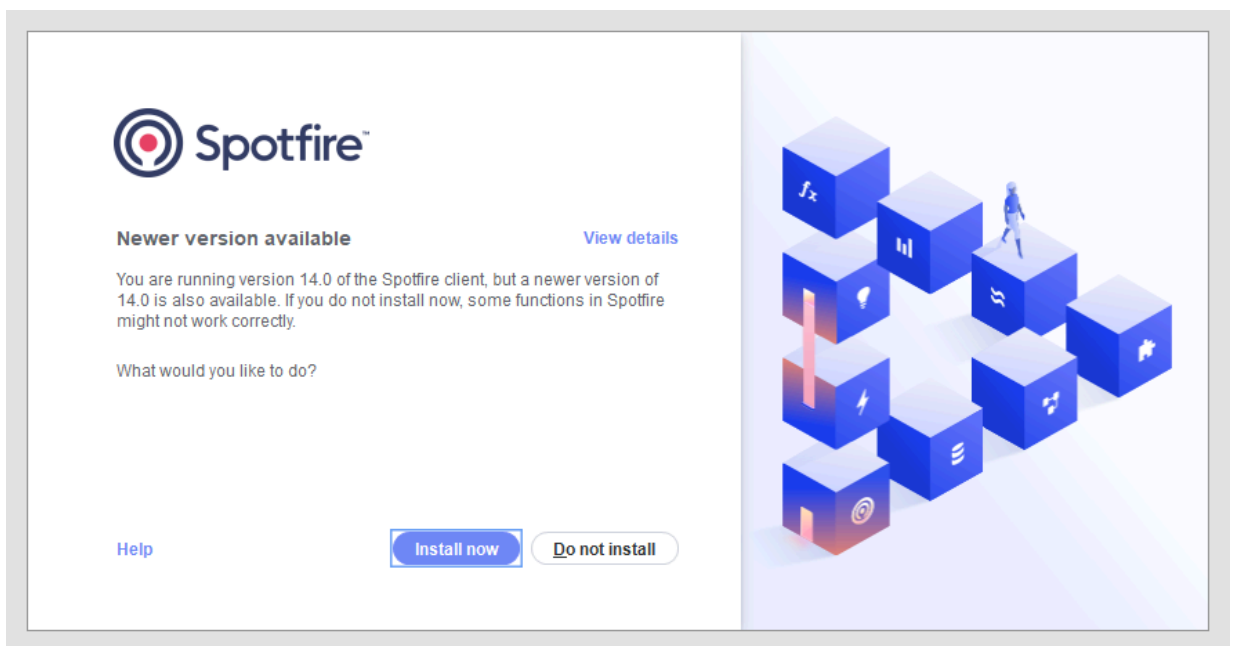


The following terms in the dialog are explained below

Option	Description
Select deployment area	<p>Allows you to select which deployment area to connect to.</p> <p>Deployment areas are configured by the Spotfire administrator and make it possible for your company to give different users access to different versions of the Spotfire client, while still using a single Spotfire server.</p> <p>Contact your administrator if you do not know which deployment area to connect to.</p>
Remember area	<p>Select this check box to always connect to the selected deployment area and skip this step of the login in the future (as long as you have access to the selected area).</p> <div>  <p>If you have selected this check box but later need to connect to a different deployment area, you can go to the <a href="#">Edit server view</a> of the login dialog and click on the link at the bottom of the dialog. This clears the preselected deployment area and allows you to see the Select deployment areas view of the login dialog again, provided that your group access still allows multiple deployment areas.</p> </div>

### Another client version available

This view of the login dialog is shown if your current version of Spotfire is not the same as the one on the server you are logging in to. Different messages will be shown depending on how your current Spotfire version differs from the version deployed on the server. Most often the version on the server is newer than the version you are running, but in some cases the version on the server might be older or differ in other ways. Contact your Spotfire administrator if you are uncertain whether to install the version on the server or not.



The following terms in the dialog are explained below:

Option	Description
View details	Click here to see the deployment packages that are different from your current Spotfire client version.

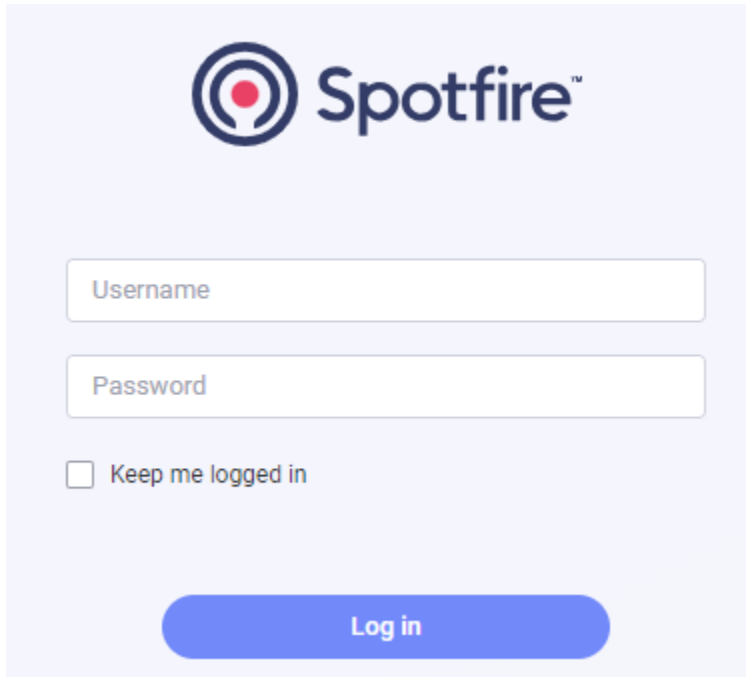


Option	Description
Install now	This will update your current version of Spotfire to the one deployed on the server.
Do not install	This will open your current version of Spotfire without installing anything.

## Logging in to the web client

To log in to the Spotfire web client, enter your username and password, and then click **Log in**.

You log in to the web client from your web browser, either by going to the server address directly, for example, `https://myspotfireserver.mycompany.com/spotfire/ui/login`, or by clicking on a direct link to an analysis from a dashboard or email, or similar.




The image shows the Spotfire login interface. At the top is the Spotfire logo, which consists of a stylized 'S' inside a circle followed by the word 'Spotfire™'. Below the logo are two input fields: 'Username' and 'Password'. Under the 'Password' field is a checkbox labeled 'Keep me logged in'. At the bottom of the form is a blue button with the text 'Log in'.

When you are running the web client, your administrator has usually assigned you or your group to use a specific deployment area or, you will use the default deployment area (often called "Production").

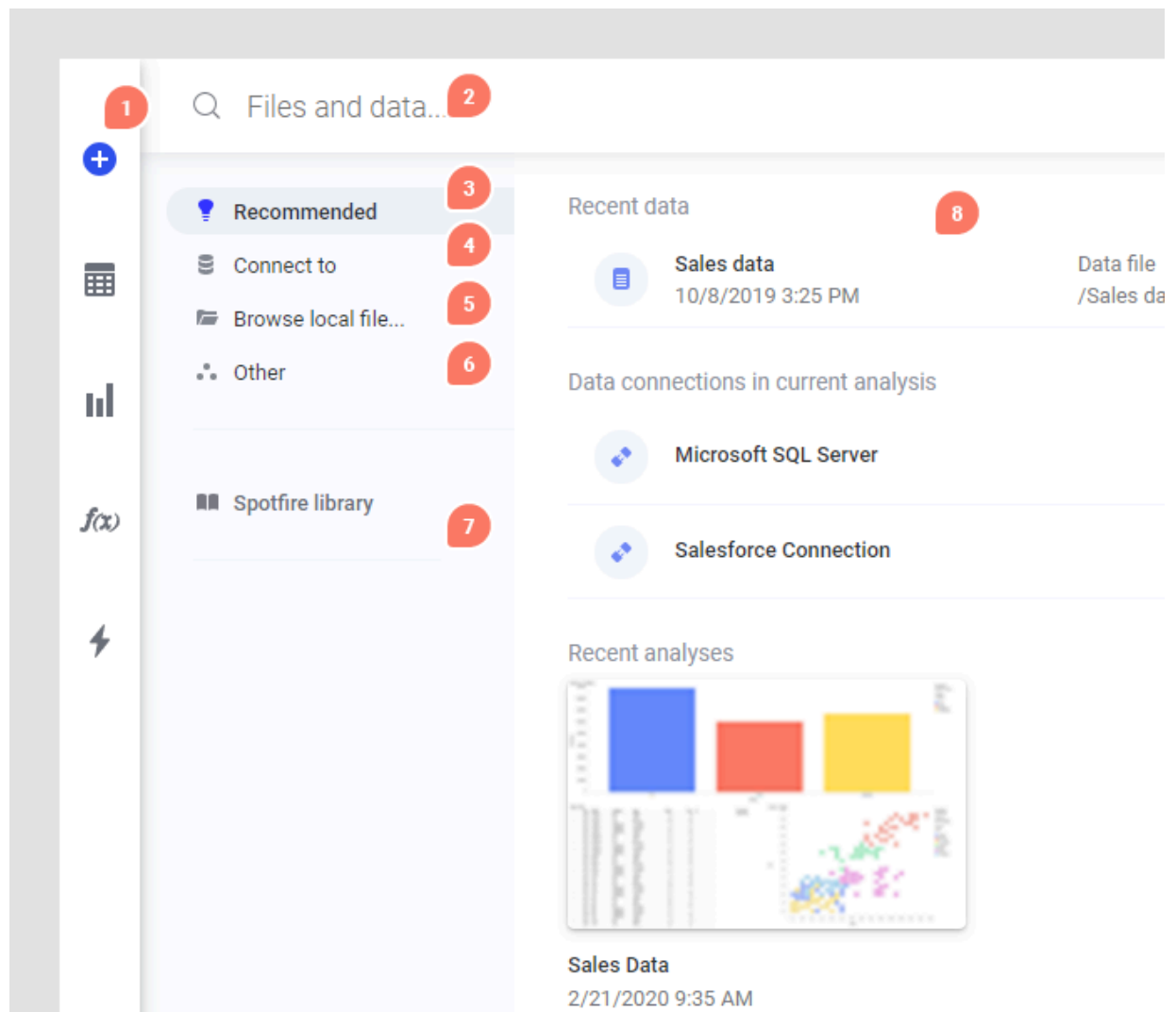
If you are an administrator user, you might have access to other sections than the **Analytics** section (with the Spotfire web client) on the server. See [Administration interface introduction](#) in the *Spotfire® Server and Environment - Installation and Administration* user guide for more information.

# Loading data

The first thing to do when you start working with Spotfire is to add some data to analyze. If you are an author, you load the data to be analyzed and then create visualizations based on it. If you are a consumer, you launch an analysis that already exists, where an author has selected the data and created visualizations. The data in an analysis can be retrieved from, for example, a Microsoft Excel spreadsheet, a text file, data saved in the library, a data function designed to load data, or, data from various databases.

You add data or open analyses via the **Files and data** flyout (reached by clicking  on the authoring bar).

The **Files and data** flyout contains a number of different elements:



1. The plus-sign is the access point to the **Files and data** flyout.
2. Search for files, data sources and data in the upper part of the flyout. You can specify what item type to search for using keywords as described in [Searching the library](#) on page 124.
3. The **Recommended** view will show different things depending on the context. For example, you may find sample data, or, recent analyses or recent data here.

- Under **Connect to**, you will find a list of all available data connectors. Click on a connector to see a list of data connections from that source in the analysis, in the library, or to create a new connection. Note that the installed client has access to authoring with many more data sources than the web client.

See [Supported data sources](#) on page 86 for more information.

- Click **Browse local file** to choose data from one or more file sources, or to open local DXP files.
- The **Other** option is sometimes available. It then contains the **Linked copy to data table in analysis** option (previously known as "data table from current analysis" or "data table data source") which can be used, for example, if you want to unpivot the data in an already existing data table but also keep the original data table.

Using the installed client, you can also **Paste data from clipboard**, open **Information designer** to create new information links, open **Manage data connections**, or open data from databases which currently do not have a native data connector (which otherwise is the preferred option) using the **Load Data with ODBC, OLE DB, or ADO.NET data provider** option.

- Click **Spotfire library** to locate folders and files by browsing the library structure.
- The right part of the flyout shows the available options for the selected category. For example, here you will find recent analyses or recent data, a list of available data connectors, files or folders in the library, or other types of access points.




You can also drag a supported file to the analysis directly and drop it in your browser or installed client.

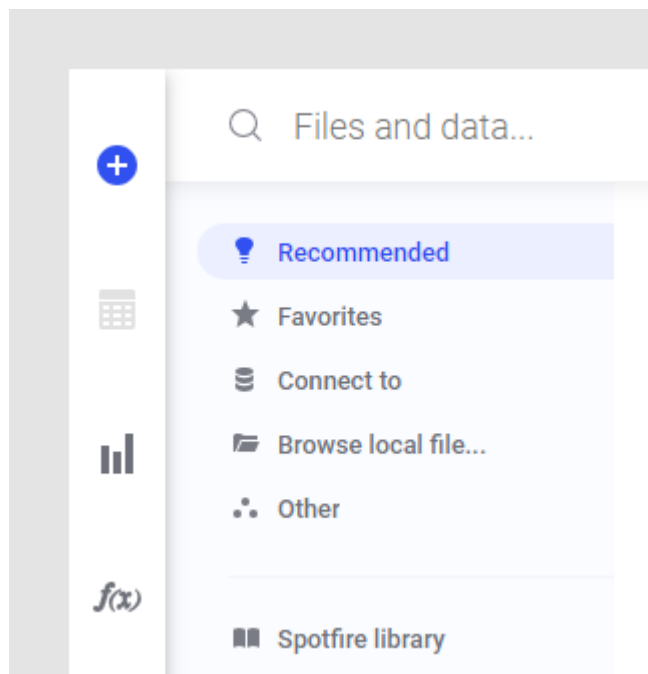
## Choosing the data to analyze

You select what data to analyze from the **Files and data** flyout.

See [Loading data](#) for more information about the different categories available in the flyout.

### Procedure

- On the [authoring bar](#), click **Files and data** .



2. In the flyout, locate the data of interest by searching for a suitable filename or keyword, or by browsing the different categories.

You can search for specific types of data by typing keywords. For example, enter `type:dxp` (to search for analyses), `type:dataconnection` (to search for data connections), or, `type:query` (to search for information links). See [Searching the library](#) for more details.

Click **Connect** to see a list of all available data connectors.

Click **Browse local file** to open files from your computer or a network location.



You can add several local data files simultaneously.

You might also see a section called **Other**, with the [Linked copy to data table in analysis](#) on page 43 section (previously known as "data table from current analysis" or "data table data source"), which can be used, for example, if you want to unpivot your data but also keep the original data table in the analysis.

In the installed client, the **Other** section also contains **Paste data from clipboard**, open Information designer, Manage data connections, and the option **Load data with ODBC, OLE DB, or ADO.NET data provider** which allows you to open data from databases that currently do not have a native data connector (which otherwise is the preferred option).

3. If a configuration dialog shows up, change any settings to your liking.  
If you open a data connection you might need to log in before you can select the actual data. For Excel files with multiple worksheets you can choose which worksheet to import. If you cannot see all worksheets available in an Excel file at this step, try saving and closing the file in Excel before you open it in Spotfire.
4. When you have selected your data, you will see a summary view of all data tables that will be added or updated. In many cases, you can make changes to the default recommendations by clicking on the sources and edit settings. See [Choosing how to load the data](#) on page 55 for more information.
5. When you are satisfied, click **OK**.

## Creating a new analysis from data in the library

You can create new analyses from data saved in the library.

Data can be saved to the library using Spotfire Analyst, in the form of Spotfire Binary Data Format files (.sbf), data connections, or information links.

### Procedure

1. In the library, locate the data of interest and click the item to add the data to a new analysis.  
Alternatively, in the library, click **New analysis**, and, in the **Files and data** flyout, click **Browse the library** and locate the data of interest. You can also use the search field at the top of the flyout to find your data.  
The selected data is added to the summary view in the flyout.
2. In the flyout, you can change the name of the data table and you may have the option to change [how data is loaded](#), or, how data should be added to the analysis. Click on the source name of the data for more options. See [Choosing the data to analyze](#) on page 39 for more information.
3. Click **OK** when you are satisfied.

### What to do next

The loaded data can be visualized either using recommendations based on selected columns in the **Data in analysis** flyout, or, you can start by selecting specific **Visualization types**, or, **Explore by searching** to find data and features.

## Creating a new analysis from a local file

You can create new analyses from files that are saved locally on your computer.

This topic describes how to create an analysis based on a Microsoft Excel Workbook, but you load data from other file formats in a similar way.

You can create new analyses from locally saved files of the following file formats: Microsoft Excel Workbooks (.xlsx, .xls), Microsoft Excel Binary Workbooks (.xlsb), Microsoft Excel Macro-Enabled Workbooks (.xlsm), comma separated values files (.csv), text files (.txt), logfiles (.log), Spotfire Text Data Format files (.stdf), Spotfire Binary Data Format files (.sbdf). You can also open locally saved Spotfire analysis files (.dxf), but then you will not create a new analysis; instead an existing analysis will be opened.



To interpret the Excel worksheet correctly, you may need to make some [data preparation](#) before loading it.

### Procedure

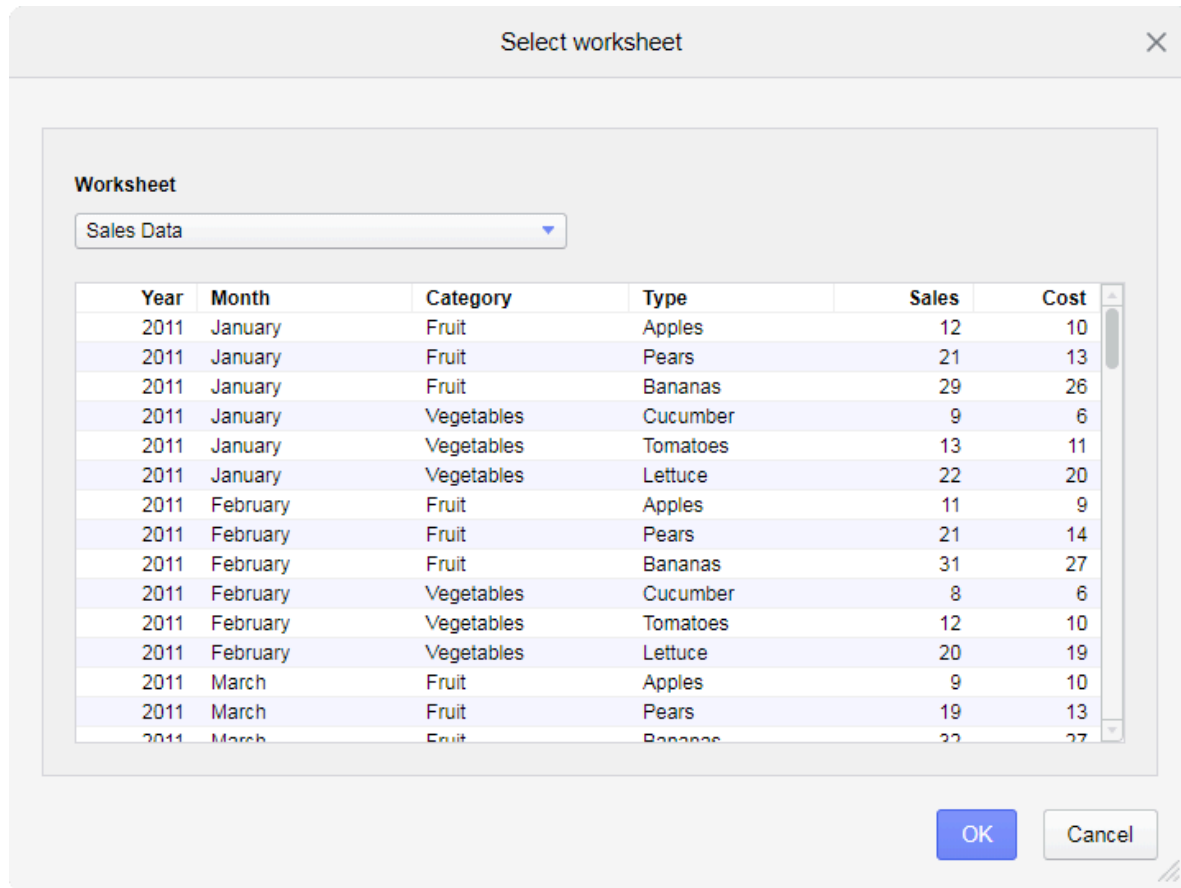
1. In the library, click **New analysis**.
2. In the **Files and data** flyout, click **Browse local file** and locate the data of interest.



You can add several local data files simultaneously.

3. Select the file, and click **Open**.

If your Excel Workbook contains more than one worksheet, the Select worksheet dialog is shown. In the dialog, select which worksheet to load, and then click **OK**. Only one worksheet can be loaded at a time.



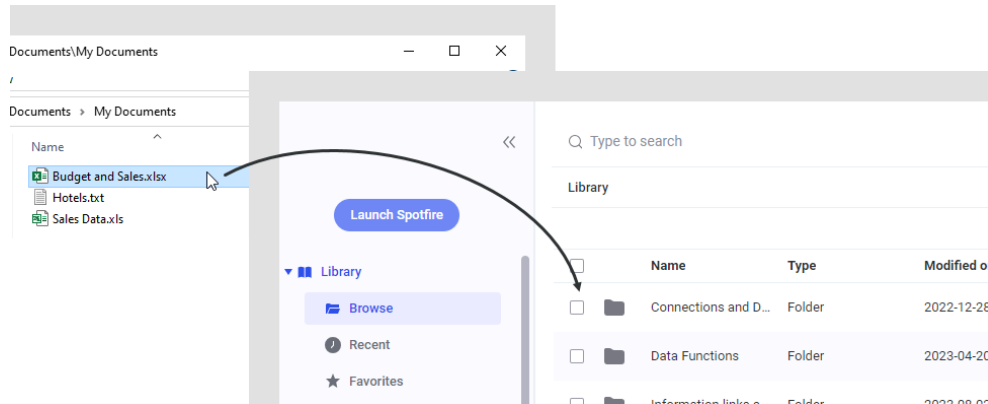
If you cannot see all worksheets at this step, try saving and closing the file in Excel before you open it in Spotfire.

The selected data is added to the summary view in the flyout.

4. In the flyout, you can change the name of the new data table. Click **OK** when you are satisfied.

## Dragging a file to the browser

Alternatively, you can add data from a local file by dragging a file to the library or to an analysis.



If the file is an Excel Workbook with more than one worksheet, select the worksheet as described above.

## What to do next

The loaded data can be visualized either using recommendations based on selected columns in the **Data in analysis** flyout, or, you can start by selecting specific **Visualization types**, or, **Explore by searching** to find data and features.

## Linked copy to data table in analysis

The option to add a linked copy to a data table in your analysis is used when you want to add data from a data table already included in the analysis to a new data table, or as rows or columns in another data table.


For example, this might be useful if you want to use both the original and a pivoted version of your data in the same analysis. You might also want to perform analyses on the data source both as a separate data table but also to include it as added rows or columns in another, bigger data table.

By adding a linked copy instead of adding the same data multiple times, you can reduce the load on an external source because you only bring in the original data once. You can also keep the file size of the analysis to a minimum if you use stored data.

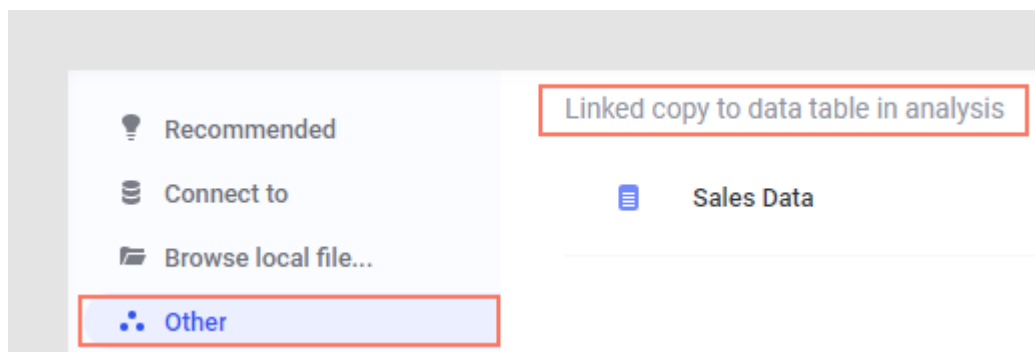
## Prerequisites

You must have added at least one data table to the analysis already.

## Procedure

1. On the [authoring bar](#), click **Files and data** .

- Click **Other** and then, under **Linked copy to data table in analysis**, select the data table to copy.



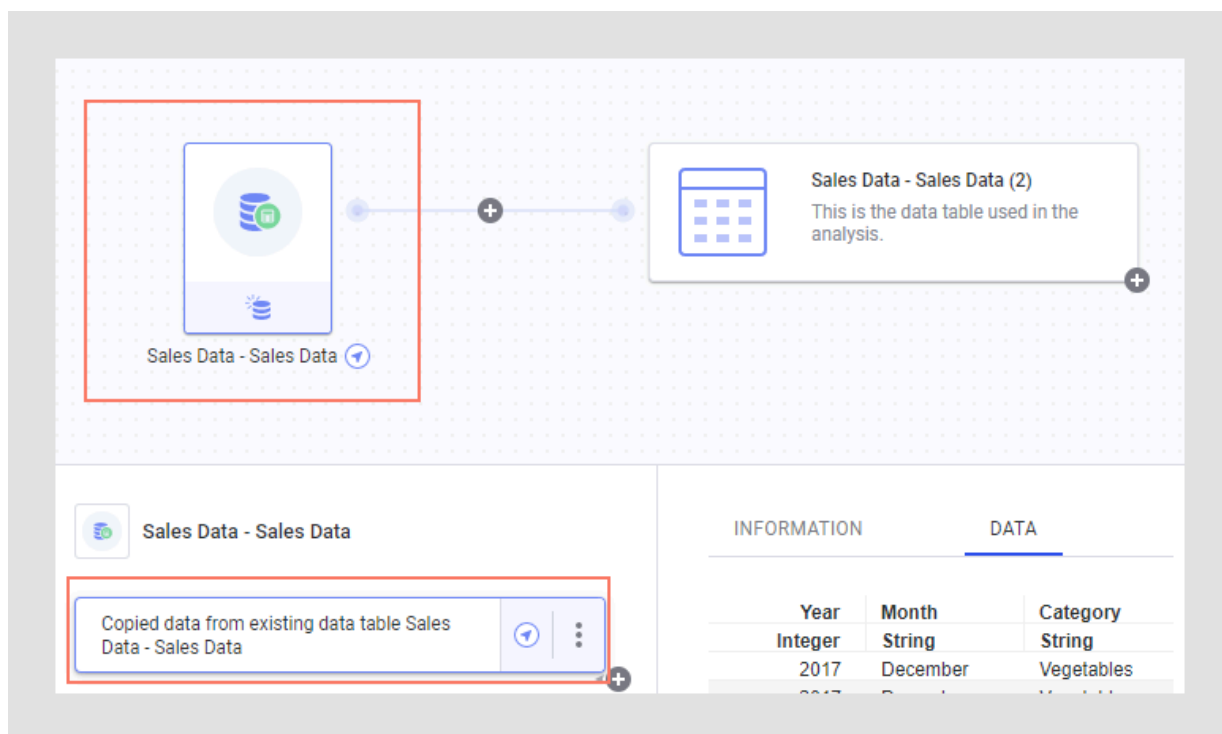
The summary view is opened and you can determine how to add the linked data. See [Choosing how to load the data](#) on page 55 for more information.

- Click **OK** when you are done.

### Result

The data is added to the analysis as specified.

You can navigate to the linked data table when inspecting the new data in the data canvas by clicking on the navigation arrow, either on the node in the graphical structure or in the step list in the lower left part of the canvas.



In all new analysis files (v 7.0 and forward), data tables based on a linked copy to data table in analysis (previously called 'Data table from current analysis' or 'data table data source') are automatically updated when the source data table changes. However, if you have old analysis files, you can update the analyses to get automatic updates there too. This can only be done using the installed client. In the step list to the lower left, select **Edit settings** from the menu on the step and then select **Automatic update** in the dialog.



## Types of data in Spotfire

Data can originate from a multitude of different sources and the trends move towards a general supply of more and more data. It can be a matter of traditional enterprise sources, relational databases, cubes, NoSQL, machine generated data, cloud, streaming data, dark data, unstructured data, big data in Hadoop, Web Services, etc. With Spotfire you have the perfect environment for analyzing any type of data, from any source.

There are a number of different ways available to access your data:

- **Direct file access** (.xls, .xlsb, .xlsx, .xlsm, .csv, .txt, .stdf, .sdbf, .shp, .geojson) and even more file types using the installed client (.mdb, .mde, .sas7bdat, .sd2, .udl, etc.). You might also have access to additional file sources, if such have been configured by your company. File data is always analyzed in-memory.
- **Data connections** via connectors to external systems. The data connections are well suited to handle big data and each connector is tailored for its particular source. Custom queries can be created in order to fine-tune the data returned or the performance of the query. With data connections, you can usually choose to analyze the data either in-memory or in-database.

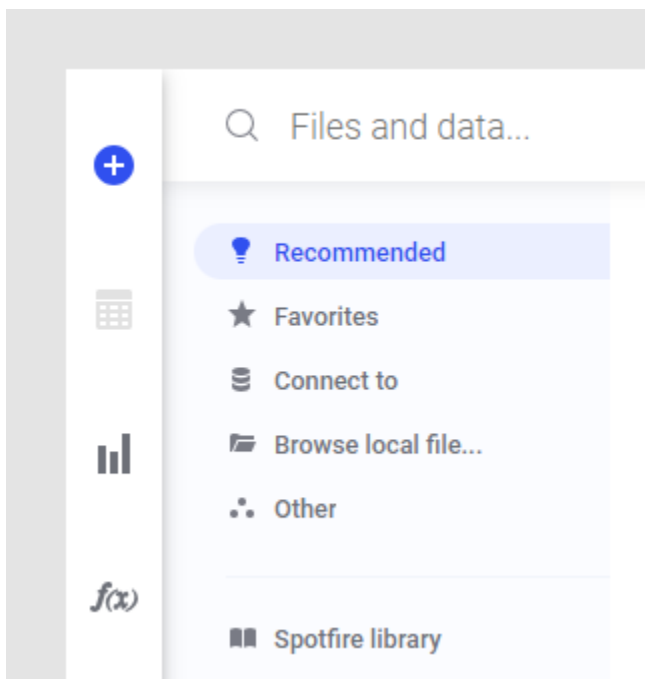
**Streaming data** is a special case of in-database data where the data is updated in real-time, or at any frequency, depending on how you configure your connection to the streaming data. Connections to streaming data work the same way as any other connection.

- Spotfire **Information Services** and its configuration tool Information Designer (available from the installed client only) allows you to set up any JDBC data source and create information links, which are predefined SQL queries that can be analyzed in-memory. Information links can include prompts to enable the end users to select their own data and procedures can be added both as pre-/post- or main queries. A built SQL query can easily be edited for further tweaking of the result.

Information links are not available in a Cloud environment.

- **TIBCO Data Virtualization** (also branded as Spotfire Advanced Data Services) is an interface to other third-party applications, commonly used for access to Web Services.

You can access all types of sources from the Files and data flyout on the authoring bar:



Data from multiple sources can be combined into a single Spotfire analysis and analyzed using a number of different visualization types. If Spotfire detects that new data is similar to previously added

data, you will get a recommendation to join your data into a single data table when adding new data to an analysis.

Data from different sources can be shown in a single visualization, or in many. Different data tables can be [related to each other](#) so that filtering in one affects the values in another data table, but they can also remain completely separate: there are no limits.

Below is an overview of the different load methods and features available for different access types. See also [Load methods](#) on page 58 for more details.

Supported?	Connectors	TIBCO Data Virtualization	Information Services
In-database	Yes	Yes	–
In-memory	Yes	Yes	Yes
On-demand	Yes	Yes	Yes
Custom queries	Yes	-	Yes

## Working with large data volumes

When you are working with massive amounts of data there can always be certain operations that take time to perform. However, with Spotfire you do not have to be afraid to try out different alternatives. You can always cancel an operation if it looks like it is going to take a long time. You can undo an operation, or switch to a different alternative (for example, switch to a column with fewer unique values on an axis) if you do not want to wait for the calculations to finish.

However, here are a few tips which can be useful when you are working with large data tables and you want to increase the performance of your analysis:

## Visualizations and analyses

- Use aggregated visualizations as a starting point and use details visualizations for smaller, filtered portions of data only. Many graphical elements in the analysis will take some time to render. This is especially important on the web client which does not allow hardware acceleration.
- Think about whether there are any alternative ways you can visualize your data in order to see the same thing. Can you use a different visualization type? Or partly aggregate the data? For example, binning can be used to aggregate markers in a scatter plot and still allow you to see a distribution. Using the bin sliders you can increase the number of markers shown until it takes too long time to make changes.
- Sorting in cross tables, etc., takes time.
- Hide or delete unused filters (or do not create filters for external columns unless you have to).
- Use the list box filter or the text filter rather than the item filter when working with columns with a lot of unique values. Item filters are costly to show, even when they are not used. If you have old analysis files using item filters for these type of columns you can manually change the filter type to a list box or text filter and save the file again.
- Some types of aggregations are more time consuming than others. For example, use average rather than median, if possible.
- Use the data type real rather than currency. The currency formatter can be applied to the real data type.
- It is recommended to use the filters panel or the data in analysis flyout instead of adding a lot of filters to text areas. Many filters in text areas can make the analysis less responsive.

- Calculated values (labels) and sparklines in text areas might also lead to less responsive analyses.
- Use post-aggregation expressions for all expressions including OVER since these calculations are faster when done on an already aggregated view.

## Hardware

- Use a fast solid-state drive (SSD) if possible (when data or analyses are stored on disk).
- Do not run other applications on the same computer when working with large data volumes.

## Loading data

- Use sorted input on categorical columns.
- Loading data from an SBDF file is much faster than from TXT.
- If the data is in a tall and skinny format rather than a short and wide you may obtain better performance.
- Remove invalid values from your data before importing into Spotfire.
- If you intend to import data from an external data source, limit the selected data as much as possible prior to import. This will increase the chances that the import is successful.

## Data export

- Export from a data table rather than from a table visualization.
- Export to SBDF rather than to TXT.

## Web client

- Avoid visualizations with many graphical elements, if possible.
- Use scheduled updates, when possible.

## Preferences (on-premises only)

- An administrator using the installed client can modify the `MarkingWhereClauseLimit` or the `MarkingInQueryLimit` preference (under **Tools > Administration manager > Preferences > DataOptimization**). With lower limits, the allowed complexity of marking queries is reduced. This is important when working with external data sources. See [Preferences Descriptions](#) in the *Administration Manager* help for more information.
- Switch off the automatic creation of filters. This can be done for a specific data table in the [data canvas Settings](#), and for all new in-memory data tables under **Tools > Options > Document** (installed client only).

## API

- Prefer iterator based data access over random access. Use `DataRowCursor` API:s over `GetValue(rowindex)` style API:s.
- Be careful when using custom comparers - depending on usage they may become a bottleneck. Consider if the problem cannot be solved in other ways.

- If things are slow and you are using old custom extensions, see if they can be refactored or if some time-consuming steps can be removed. Some APIs are by nature slow and old code might benefit from some refactoring. Try loading without any extensions to see if one of them may be the culprit.

## Working with in-database data

When you are working with data from an external data source (in-database or in-db data) there are a number of differences to keep in mind, compared to working with data in memory. Some features work differently or are unavailable.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

If you are working with streaming data, there are some additional differences to keep in mind, see [Working with streaming data](#) on page 52 to learn more.



One thing to think about when working with in-db data is that changes to the underlying database schema will not automatically be reflected in the Spotfire analysis. This means that if a column is added to a database table you must perform a **Refresh Schema** operation in Spotfire to see the new column in the analysis. Not all users will have the sufficient database privileges to perform a full schema refresh. However, changes to the actual data can be updated using a simple Reload/Refresh by most users. See [Reloading data](#) on page 227 for more information.




Before you can work with in-db data in Spotfire there are few prerequisites that must be met (for details, see [Spotfire® System Requirements](#)):



- You might need drivers for the data source of interest installed on your computer.
- You must have been granted access to the licenses for all connectors of interest by your Spotfire administrator.

Difference with in-memory	Why?
The table visualization and the Details-on-Demand can only display a limited number of rows (10 000 by default) from a relational in-db data source.	There is a preference setting limiting the maximum number of rows that can be shown in table visualizations and Details-on-Demand, to prevent large databases from overflowing Spotfire.
Virtual columns are not supported for in-db data.	The default limit is 10 000. A Spotfire administrator can configure the limit by editing the preference setting <code>TableVisualizationExternalRowLimit</code> in the Administration Manager.
A Spotfire administrator might configure a limit for how many rows can be retrieved and visualized, when working with in-database data. This limit affects, for example, visualizations and certain filter types.	When exploring large data tables, there is a risk of loading very large amounts of data. This can result in poor performance, because of the strain put on the network, the database and the system memory, particularly when using the web client.
Trying to visualize data in a way that exceeds this limit, for example by not aggregating the data enough, will result in an error.	To avoid these problems, there is a preference setting that can be used to limit the number of rows of data that can be retrieved and visualized, when working with in-database data.
List box filters, hierarchy filters and item filters cannot be created for variables that contain too many unique values.	A Spotfire administrator can configure the limit with the preference setting <code>ConnectorExternalRowLimit</code> in the Administration Manager.
In the details view of the expanded <b>Data in analysis</b> flyout, you cannot see statistics for categorical columns that contain too many unique values.	

Difference with in-memory	Why?
<p>For in-db data from cubes you must select the columns to include in table visualizations manually.</p> <div>  <p>You can enable this behavior for in-database data tables from connections to non-cube sources as well. Use the setting <b>Manually add columns for table visualizations</b>, available on the <b>Performance</b> tab of the Data Connection Settings dialog (installed client only).</p> </div>	<p>Using all available columns in a table visualization can result in too large queries towards the OLAP source, because the table view flattens the cube into a single table.</p> <p>By selecting only the most important columns and perhaps use table visualizations as details visualizations only, you can reduce the impact of queries from Spotfire against the data source.</p>
<p>The table visualization cannot be sorted if the data table source is a cube.</p>	<p>Cubes can only return data in the natural order and reversed natural order, not sorted orders. Therefore, sorting is disabled for all table visualizations based on cube data.</p>
<p>There is a limit on markings in a visualization, when working with in-db data.</p> <p>Because marking of more items (for example, bar segments, pie sectors or markers) results in a more complex query to the external data source, the limit will affect the number of items you can mark. Different visualization types and configurations have varied complexity, so the number of items you can mark will differ between visualizations.</p> <p>The default value of the limit is 1000. This preference setting can be configured by a Spotfire administrator.</p> <div>  <p>If you want to include more items in a marking, you can add items to an existing marking by holding Ctrl or Shift and marking further items. This way, you can create markings exceeding the marking limit.</p> </div> <div>  <p>Marking all items in a visualization will not result in a complex query. This means that all items can always be marked, no matter what the limit is.</p> </div>	<p>When you mark items in a visualization, Spotfire will generate a query to the external data source. Such a query can easily become very long and complex, if a visualization contains a large number of items; especially if you also have filtered values in the analysis.</p> <p>Complex queries can cause poor performance. Therefore, a preference setting is available, limiting the number of items that can be marked in a visualization when working with in-db data.</p>
<p>Marking in table visualizations is only supported if the view has a primary key defined. This can be defined by an administrator in the underlying database table, or, for some types of data connections, in the Views in Connection dialog (installed client only). You cannot mark in tables based on cube data.</p>	<p>A primary key is required to know which rows to mark.</p>

Difference with in-memory	Why?
<p>Marking is disabled in visualizations which you have configured with a floating point expression on a categorical axis.</p> <p>If you want to use marking in a visualization that is based on in-database data, avoid using categorical expressions (written between angle brackets, '&lt;&gt;') that return floating point values, on any axes.</p> <div data-bbox="279 527 323 569">  </div> <div data-bbox="391 432 821 667"> <p>It is not only when you use custom expressions that you might encounter this limitation. For example, if you select a floating point column on the category axis of a bar chart, and configure the category axis to show a categorical scale, this is the same as writing a categorical custom expression that returns a floating point value on that axis.</p> </div>	<p>Marking in in-database visualizations, which you have configured with a floating point expression on a categorical axis, can produce unpredictable results.</p>
<p>The summary table cannot be used with in-db data.</p>	<p>The summary table has not yet been adjusted to be able to handle in-db data.</p>
<p>The box plot cannot be used with in-db data.</p>	<p>The box plot requires both aggregated and non-aggregated data and has not yet been adjusted to handle in-db data.</p>
<p>Scatter plots and 3D scatter plots might need to be configured as aggregated visualizations when using in-db data.</p>	<p>The number of rows available on an external data source might be too large to handle within Spotfire.</p>
<p>Dendrograms can only be shown in heat maps if the X-axis is set to (None), when using in-db data. Highlighting and marking in the dendrogram is not supported.</p>	<p>Dendrograms are row-index based and in-db data tables do not have any row index.</p>
<p>Not all of the standard Spotfire aggregation methods and expressions are available on all external systems. On the other hand, there can be other methods available that Spotfire does not have by default.</p>	<p>The aggregation methods supported by the external data source determines which methods will be available for in-db data.</p>
<p>OVER expressions must be written as post-aggregation expressions, or as custom expressions, where applicable.</p>	<p>OVER expressions must be performed on the aggregated view. See <a href="#">Using expressions on aggregated data (the THEN keyword)</a> on page 777 for more information.</p>
<p>Automatically created date and time hierarchies are not available for in-db data.</p>	<p>The shortcuts for setting up date and time hierarchies have not been adjusted to handle in-db data.</p>
<p>Transformations are not available for in-db data.</p>	<p>Because no data is stored within the analysis in runtime, it is not possible to apply any transformations on in-db data.</p> <p>However, data tables based on data connection views can also be imported and once the data is in-memory, you can apply transformations to it. See <a href="#">Transforming data</a> in the Spotfire Analyst User Guide for more information.</p>
<p>If the connection to the external data source is lost, no further analysis using that data source can be performed.</p>	<p>Because no external data is stored within the analysis in runtime and each change induces a new query to be sent to the external data source, the data source must be available as long as changes to the visualizations are to take place.</p>
<p>Tags are not available for in-db data.</p>	<p>Tags are dependent on row numbers which are not available for in-db data.</p>

Difference with in-memory	Why?
<p>You cannot delete columns or rows from in-db data tables.</p>	<p>In-db data tables contain no data in runtime so nothing can be deleted.</p> <p>However, you can hide columns from the source tables when selecting tables in the Views in Connection dialog (for relational data sources) or the Data Selection in Connection dialog (for cube data sources) using the installed client.</p>
<p>You cannot apply any of the following tools or operations on in-db data tables:</p> <p>Add columns,</p> <p>Add rows,</p> <p>Add binned column,</p> <p>Data relationships,</p> <p>K-means clustering,</p> <p>Line similarity,</p> <p>Data functions,</p> <p>Regression modeling,</p> <p>Classification modeling,</p> <p>Add predicted columns</p>	<p>In-db data tables contain no data in runtime and can, hence, not incorporate any other data.</p>
<p>Add calculated column has the following limitations when working with in-db data:</p> <ul style="list-style-type: none"> <li>• No cube connectors are supported.</li> <li>• Some connectors (e.g., Cloudera, Hortonworks and Vertica) do not support aggregation methods.</li> <li>• Binning is not supported.</li> <li>• Post-aggregation expressions (the THEN keyword) are not supported, which means that OVER expressions are not supported.</li> </ul>	<p>The expression for the calculated column is evaluated in the underlying database management system. This means that only methods supported by the current connector will be available.</p>
<p>Cube attribute or dimension columns cannot be used on numeric axes.</p>	<p>Since all cube dimensions are mapped as String when cube data is added to Spotfire, these columns cannot be used on any numeric axes even if the name of the column may indicate that the result is numeric.</p> <p>By importing parts of the cube data it is possible to convert string data to other data types.</p>
<p>On-demand on external data is not available directly when first adding the data connection to an analysis. Instead, you must add the data as <b>External</b> data and edit the On-demand settings from the data canvas (installed client only) to enable on-demand loading.</p>	<p>The Add data to analysis workflow always assumes that on-demand data should be imported.</p>
<p>On-demand is currently not supported for cube data sources (neither for imported cubes nor for in-database cubes).</p>	<p>On-demand has not yet been adjusted to be able to handle cube data.</p>
<p>In contrast to data connections from relational data sources, selections for cube data connections always result in a single data table.</p>	<p>The data selection steps for cube data sources are currently focused on limiting the flattened cube data, not on viewing different slices simultaneously.</p>

Difference with in-memory	Why?
Auto-bin Column only works for certain connectors; for example Teradata, Oracle, PostgreSQL, and SAP HANA.	The ability to use width_buckets for auto-binning is only supported by some external data sources.
Export Data from Data Table cannot be used when exporting from in-db data tables.	<p>In-db data tables contain no data in runtime so nothing can be exported from the data table.</p> <div>  <p>Use the <a href="#">Export data from Visualization</a> option when you want to export from in-db data tables. Note that for all visualizations other than the table and the cross table you must mark the items of interest before opening the Export data dialog.</p> </div>
String comparisons between different systems may provide different results due to handling of trailing spaces.	External systems that adhere to the SQL-92 standard (for example, SQL Server, MySQL and Netezza) will trim trailing spaces for strings.
You must provide wildcards to obtain fuzzy matching in text filters when working with in-db data.	Typing a part of a word with in-db data will only search for the exact match of those letters. (With in-memory data there will automatically be a wildcard appended to the entered letters.)
<p>By default, if you have multiple filters for an in-db data table, the values that are available for selection in a filter do not reflect any filter selections you might have already made in other filters. This means that values that might already be filtered out will be available for selection, and you will not get the visual clue indicating which values have been filtered out using the other filters, the way you do with in-memory data tables.</p> <p>To make filters for an in-database data table work more like filters for in-memory data tables, you can change the settings for the data connection to enable cascading filters for in-database tables.</p>	<p>For in-database data tables, hiding and graying out values that have been filtered out by other filters is a costly operation. Therefore this option, called cascading filters, must be enabled manually.</p> <div>  <p><b>CAUTION</b>Enabling the cascading behavior for filters for in-database data tables results in an additional query load on the external data source. It is the user's responsibility to ensure that the external data source can handle the extra query load.</p> </div> <p>To enable cascading filters for in-database data tables from a connection:</p> <ol style="list-style-type: none"> <li>1. On the menu bar, select Data &gt; Data Connection Properties.</li> <li>2. In the Data Connection Properties dialog, select the data connection of interest and click Settings....</li> <li>3. In the Data Connection Settings dialog, click the Performance Settings tab, and then select Enable cascading filters for in-database data tables from this connection.</li> <li>4. To save your changes and close the dialog, click OK.</li> </ol>

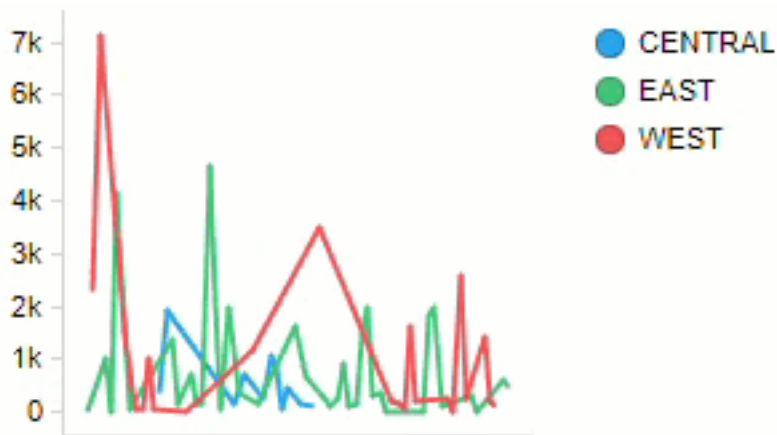
## Working with streaming data

You can connect to your streaming data in Spotfire if you want to visualize and keep track of updates in real-time.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.





Adding a connection to streaming data is not different from connecting to other in-database data. To learn more about connecting to streaming data in Spotfire Data Streams, see the [Spotfire Analyst User Guide](#). When you have connected to your streaming data, it works in many ways the same as other in-database data, but some differences are worth keeping in mind. When working with streaming data, you have yet another option to change the data limiting for a visualization, and you can choose to only show a specified time range at a time. See [Adding a streaming time range to a visualization](#) on page 564 for more information.

Before going into what differs between working with streaming and non-streaming data in Spotfire, it is also valuable to know that a visualization can be streaming for several reasons, and some differences in functionality depend on what makes the data stream. These four cases can make a visualization show streaming data.

1. The main data table in the visualization is a streaming data table.
2. A secondary data table is a streaming data table, while the main data table is an in-memory data table, or a non-streaming in-database data table.
3. The visualization uses a [marking](#) that is based on a streaming data table.
4. The visualization is a [details visualization](#) that is limited by a marking that is based on a streaming data table.



Check the status bar for the active visualization if you are unsure whether the visualization is based on a streaming main data table or not.

Note also that any visualization or feature that is unavailable or has limited functionality for in-database data in general, is also unavailable or has limited functionality for streaming data. For general information about external data, see [Working with in-database data](#) on page 48.

### Unavailable for all streaming data

The following visualizations and features are unavailable for streaming data, no matter which one of the above listed cases makes the data in the visualization stream.

- **3D scatter plots** are not available for streaming data.
- **Graphical tables** are not available for streaming data. Note however that individual dynamic items (sparklines, bullet graphs, icons, and calculated values) are available for streaming data in text areas.
- **Hierarchy filters** cannot be created from columns in streaming data tables.
- **Dendrograms** cannot be added to heat maps that are based on streaming data.

## Unavailable when main data table is streaming

The following visualizations and features are unavailable when the main data table in a visualization is streaming, but are available for the other three cases of streaming data listed above.

- **Pie charts** are not available when the main data table is streaming.
- **Pie markers in scatter plots and map charts** cannot be used when the main data table in the visualization is streaming.
- **Multiple subsets** cannot be used when the main data table in the visualization is streaming.

## Features with limited functionality for streaming data

- **Map charts** - Automatic geocoding is not available for streaming data. To perform geocoding for a column manually, click the column of interest in the **Data in analysis** flyout, make sure **Details on selected column**  is shown in the expanded flyout, and specify geocoding by clicking **Settings** and making your choice.
- **Range filters** - Columns containing string values cannot be used in range filters with streaming data.
- **Cross tables** - When the main data table is streaming, grand totals and subtotals cannot be calculated on underlying row values. You can choose to [calculate totals as the sum of cell values](#) instead.
- **Table visualizations and Details-on-Demand** - By default, a maximum of 100 rows can be displayed in table visualizations with streaming data. A Spotfire administrator can configure this limit, by editing the preference setting `TableVisualizationStreamingDataRowLimit` in the Administration Manager.
- **Export** - When exporting visualizations based on streaming data, active page is supported, while other export options have limitations.
- **Categorical axes** cannot be configured to evaluate axis expression on 'all data', but they can be evaluated on 'current filtering only' instead.
- **Lines and curves** - Some lines and curves calculate statistics on underlying data and therefore cannot be used with streaming data. For example, it is not possible to calculate  $\text{Avg}(x)$  on a bar chart with a continuous category axis.
- **Details visualizations** - When the main visualization is based on streaming data, [details visualization](#) in multiple steps have limitations: If the main visualization and the details visualization are based on different data tables with a relation set up between them, then details visualizations on multiple levels are not supported.
- **Property controls** in the text area cannot list column values from a streaming data table.
- **Filtering in related data tables** cannot be used to propagate filtering based on streaming data to another data table. However, you can propagate filtering from a non-streaming data table to a streaming data table.
- **Some configuration block settings** that depend on data being available will not work. For example, the 'uncheck all' setting for check box filters cannot be used. See [Create a configuration block](#) on the [Community](#) for more information about configuration blocks.



If you are preparing an analysis that will be used in web clients, WebSockets must be enabled on the Spotfire Server running the web client. You can open the analysis in a web client and verify that the tooltip on the globe icon reads 'Connection: WebSockets'. If it does not, WebSockets might not be enabled. Contact your Spotfire administrator for more information.

## Working with the external data row limit

A Spotfire administrator might configure a limit for how many rows can be retrieved and visualized, when working with in-database data. This is a way to prevent loading of very large amounts of data, that can strain the network, the database and the system memory. The limit affects visualizations, as well as list box filters, hierarchy filters and item filters.

When the external data row limit is exceeded you will see an error message. There are different ways that you can reduce the amount of data loaded, that might help you create your analysis without exceeding the limit. If none of the solutions apply, contact your Spotfire administrator.

See also [Working with in-database data](#) on page 48.

### Visualizations

To reduce the amount of data loaded from the database, configure the visualization to show fewer items (for example, fewer bar segments, pie sectors, lines, or markers). This can be done by avoiding to use columns with many unique values on the axes. Trellising and coloring are examples of features that can generate many items in a visualization, depending on the columns used.

[Filtering the data](#) can also reduce the amount of data that must be loaded from the database.

[On-demand loading](#) (configured using the installed client) of the data table can also be beneficial when you have access to massive amounts of data.


### Filters

The list box, hierarchy and item filter types load and display all unique values in a selected column. This might result in too much data. If you want to filter by a column with many unique values, use a filter type that does not require loading of all values. A good alternative is a text filter, which lets you type the data values to filter to.

To change the filter type, right-click the filter and, under **Filter type**, select the desired option.

### Data in analysis flyout - Details on selected column

In the [Data in analysis](#) flyout, you can see an overview of the contents and statistics for a selected column. To show this information, click to select a column in the **Data in analysis** flyout, and, in the

expanded flyout, click **Details on selected column** .

If the column contains too many unique values, statistics cannot be calculated and shown. Other overview information for the column will still be shown. For hierarchies that contain too much data, however, no overview information will be displayed.

## Choosing how to load the data

When you have selected your data and added it to the **Files and data** flyout, you will see a summary of all data tables that will be added or updated. The summary view allows you to modify data loading settings (when this is applicable), and you can also select to exclude a data source that you do not need (for example, from a predefined data connection with multiple views).

Make changes to settings in the summary view by following the steps below.

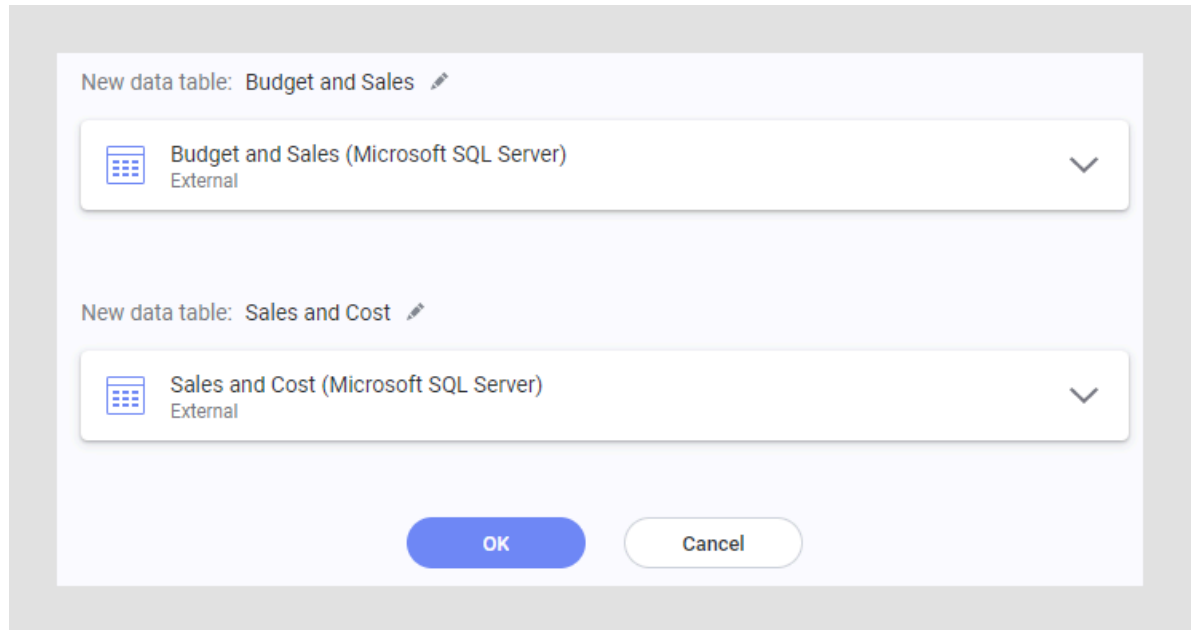
### Prerequisites

See [Choosing the data to analyze](#) on page 39 for information about how to get to the summary view.

## Procedure

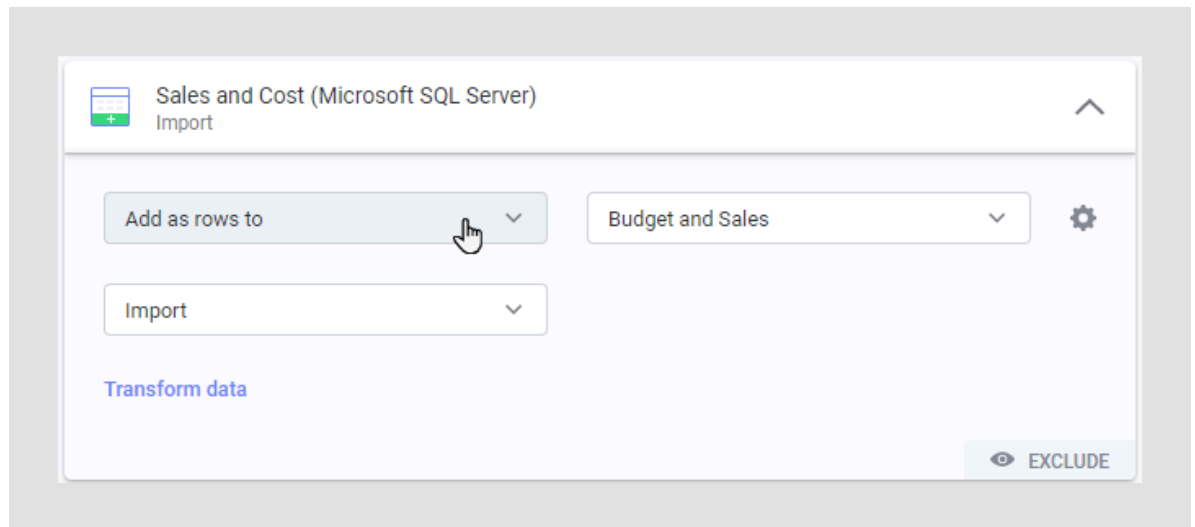
1. In the summary view, you will see one box per new source, and, depending on whether the new sources will be added as new data tables or combined with other data, there will be a header for each new or updated data table.

In this example, data was added from a data connection, containing two views. Because these two sources were not that similar to each other (and the default is to add connector data as external data), the first suggestion is to add them as two separate data tables.




2. Click on the box for a certain source to expand the box and see the available settings.
3. For data from data connections, you have the option to change the load method for the added source data. Click the drop-down list showing **Import** or **External** to change the setting. See [Load methods](#) on page 58 for more information.
4. If you have more than one data source in your analysis, and the data is imported, you can select to add new data either as a new data table or as rows or columns appended to other data. Change how

to add the selected data using the drop-down lists. If the source is a data function, you might also have access to [other options](#).



If you choose to add the new data as rows or columns, you must also select to which other data to append your data. Other choices might require more selections.

5. If desired, you can review or edit the settings for added rows or columns by clicking the settings button .

See [Editing column matching or other settings for added rows](#) on page 184 and [Editing settings for added columns](#) on page 189 for details about what you can change.

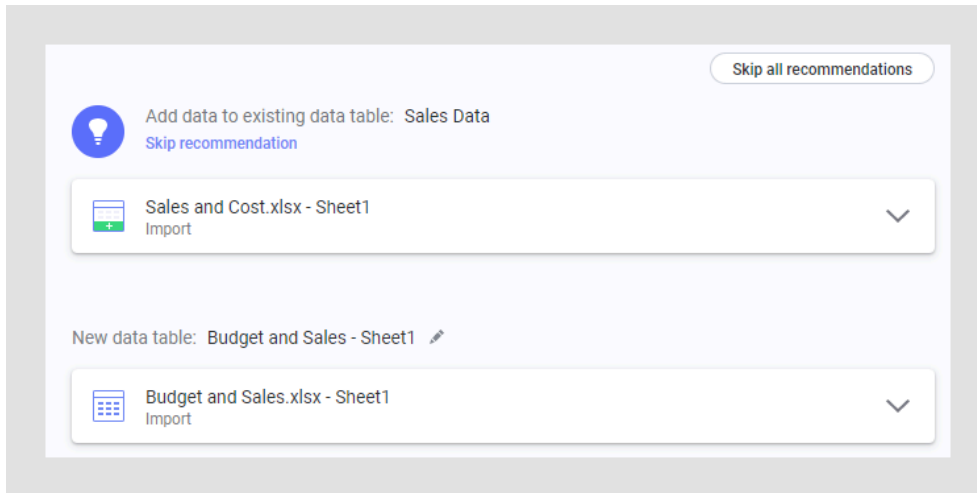
6. When working in the installed client, you might also have the option to [transform](#) data. Transformations cannot be applied to external (in-db) data connections, but you can often select to import the data from a connection and then transform the data, if the data needs modification.
7. If a predefined data connection or data function contains more views or outputs than you are currently interested in, you can select to exclude the view or output by clicking **Exclude**. Click the box again to include an excluded item, if you excluded it by mistake. You can add a previously excluded data connection view at a later stage by adding the view using the **Data connections in current analysis** option in the **Files and data** flyout (under **Recommended**), or by editing the data connection from the data canvas.
8. If desired, you can change the name of a data table by clicking the icon with a pencil next to the data table name and typing a new name.
9. When you are satisfied, click **OK**.  
The data is added to the analysis in the specified way.

You can edit settings for your data source, data function, or for added rows or columns, in the [data canvas](#) at a later stage. However, to change the load method from external to imported or vice versa, you must replace the entire data table.

If the data was added as new data tables, determine whether the new data tables should be related to each other or to previously added data tables. Remember that you must define a relation if the new data table is to be used to create [details visualizations](#) for the previously added data tables. Read more in [Related data tables, joins and column matches](#) on page 74.

## Recommendations on how to add data

If you add data from multiple imported sources to the analysis, you might see recommendations on how to add data, indicated with the light-bulb icon:



To ignore the recommendation and configure all new sources manually, click **Skip recommendation** (to skip a single recommendation only) or **Skip all recommendations** (to skip all recommendations on this page).

## Load methods

If your data comes from a data connection to an external system, you have the opportunity to choose how the data should be loaded when adding data; either as in-memory data, analyzed by the internal data engine of Spotfire, or, as in-database data (in-db data), where all calculations are handled by the external system.

See also [Types of data in Spotfire](#) on page 45.

### In-memory analysis (Import)

Text files, Excel files, and information links (not available on Cloud) are always imported and analyzed in memory, whereas, with data connections, you can choose to import the data, if desired. When you are working with data in memory, you have access to all the functionality of Spotfire, via the built-in data engine. The internal data engine is available to all users, either in the Windows client or, for web client users, on the server. You have the opportunity to use all columns as filters and you can perform many types of calculations. With imported data you can also combine data from different sources into a single data table using the [Add rows](#) or [Add columns](#) operations (and, using Spotfire Analyst, it is possible to add [transformations](#) to the data).

If your data is small enough, imported data is most of the times the preferred option, because it often improves the performance of calculations.

### In-database analysis (External)

If you choose to keep a data connection external, all calculations are done using the external system and not with the Spotfire data engine. This allows you to work with data volumes too large to fit into primary memory and take advantage of the power of the external system. When working with external data connections, you access only the current selection of data and all aggregations and calculations are made in database (in-db).

When a visualization uses in-db data, the visualization queries the external data source directly. Every time a change is made to the configuration of the visualization, e.g., a measure is defined on the Y-axis or a categorical column is added, a new query is sent to the external data source resulting in new, aggregated data.

When working with in-db data it is the connector and the underlying data source that determines which aggregation methods are available.

In-db data is usually preferred if you are working with very large data volumes, which would not fit in-memory, or if you want to make sure that the data is always the latest data from your external system, and always handled in the ways of your external system.

## Data loading settings

For in-memory data, you can specify **Data loading** settings for each source in your data table, as long as the data is kept [linked](#). These settings determine whether to use **Stored data**, **Always new data** or **New data when possible**. When saving an analysis, it is important to consider the data loading settings because they can determine whether other users will have access to the data in the analysis, if the analysis is shared.

- The **Stored data** option saves the current data in the analysis. New data will only be loaded if the source is manually reloaded.
- The **New data when possible** option will also store the current data in the analysis. In that case, data will also be available to users who do not have access to the source. However, if a user has access, then new data will be loaded when the analysis is opened.
- **Always new data** does not store any data in the analysis file.

You can change the data loading settings for applicable sources from within the **Data canvas**. See [Storing data within the analysis](#) on page 193 for more information.

If the analysis will be used with scheduled updates, you can exclude certain data sources from that update in the data loading settings. See [Reloading data for each user when using scheduled updates](#) on page 71 for more information.

See also [Reloading data](#) on page 227.

## On-demand (configured using installed client only)

When data from a non-cube data connection or an information link is to be added to the analysis as a new data table, in the installed client, you have the option to either load all data at once, or to [load data on demand](#) only. This is applicable to both in-memory and in-database data (for data connections). Your analysis can benefit from on-demand loading when you have access to massive amounts of data, but you only need to work with some parts of the data at a time. When setting up an on-demand data table you can specify conditions based on one or more other data tables to control what to load. You can also start by letting an on-demand data table be the first (or only) data table in the analysis if its input is defined by a document property, a variable you can define yourself.

On-demand loading can be regarded as a way to filter data; it is basically a WHERE-clause which dynamically limits what is read and used in calculations.

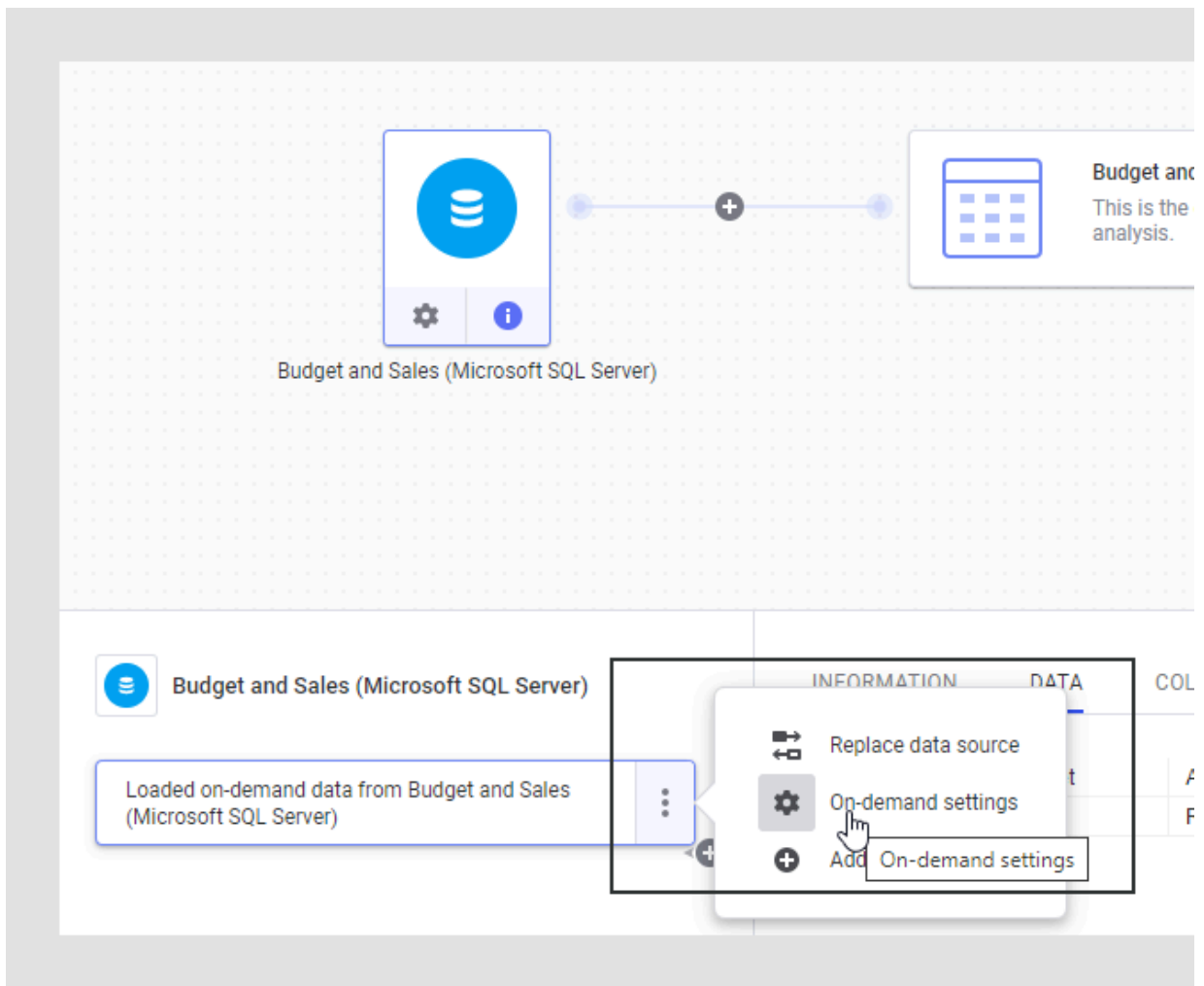
## Custom queries (configured using installed client only)

When working with data connections to relational databases or other non-cube data sources, you get the option to select one or more tables from the data source in the modeling view. Here, you might also have the option to create your own custom database query, depending on your licenses. A custom query results in a custom table which in turn can be used to set up a view in the selected connection in the same way as you would do with other database tables. Queries are written in a language that the









## Configuring an on-demand data table

When working in the installed client, data from a data connection or from an information link can be added as an on-demand data table. By editing the on-demand settings in the data canvas, you can also configure a data table from an external data connection to use on-demand.

### Prerequisites

On-demand data must be authored in the installed client.

### Procedure


1. On the [authoring bar](#), click **Files and data** .
2. In the flyout, locate the data connection or information link of interest by searching for a suitable keyword or by browsing the different categories.
3. In the summary view showing the data tables to add or update, click on the data of interest and select **On-demand** from the drop-down list showing load methods.
4. Click **Configure**.
5. Define what type of input will control the on-demand loading. In the **Define input for parameters that should control loading** list, click each of the columns or parameters from the on-demand data

table that you want to use (it might be just a single column), and specify a condition that must be fulfilled for any data to be loaded.

For each parameter that should control loading, click **Define Input** and specify how the input should be defined. For example, if you want the input to be marked values from a column in another data table in the analysis, you would pick **Values from column**, and choose the data table and column to get the marked rows from. See more details in the examples below and on [Conditions to control on-demand data](#) on page 68.

Any required prompts or parameters that were specified upon the creation of the on-demand data table will be listed as **Required** parameters in this field. This means that you must specify input handling of these parameters to load any on-demand data at all.

6. Determine whether the data table should **Load automatically** on change of inputs, or to use a manual recalculation.

If the check box is cleared, you must manually click the refresh icon on the visualization title bar to get an updated visualization.

7. Determine whether to **Allow caching**.

Allowing caching might speed up the process when loading new subsets of data. However, if the underlying data is updated during the current Spotfire session you can end up with different results for a specific set of input values depending on whether the current selection is stored in the cache. Always clear the check box if you know that the underlying data can be updated during your current session.



This caching is done on the client and it is based entirely on the *input* values, without any connection to the underlying data. No query is sent to the data source to see if that data has changed.

So, if you have defined an input based on the values in a column, data will not be reloaded unless the input has changed since the last time. If you are disabling and enabling this setting again, the existing client cache will be used once again.

8. When you are done specifying your parameters, click **OK**.
9. In the summary view, click **OK**.



In the installed client, you can edit the on-demand settings from the [Data canvas](#). By editing the on-demand settings you can enable on-demand loading for a data source that previously did not use on demand, also for an external data connection.



You might also want to add a relation between the two data tables, so the marked rows from the main data table also become marked in the on-demand data table visualizations. This can be done from the [Relations overview](#) in the data canvas.

### **An on-demand details visualization controlled by marking**


This is an example where you select to load data on demand, and you specify a marking that should control what data to be loaded.

When you mark items in a visualization that uses the specified marking, the data for the on-demand data table is updated. The update can either be done automatically each time you change the marking, or manually by clicking on the refresh button shown when the marking is changed.

This way, you can create a main visualization in which to specify the item of interest, and an on-demand visualization where more information about the marked item is loaded from the database only when requested. This is similar to the use of a [details visualization](#) based on external data.

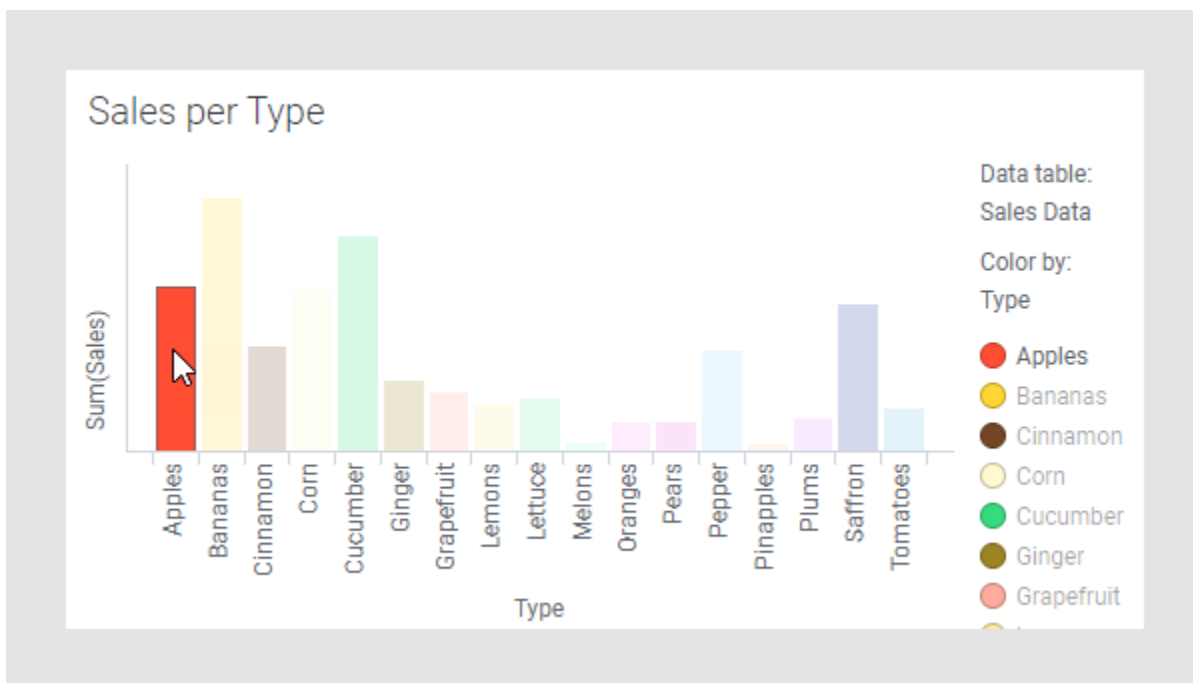
When one identifier column in the first data table is matched by a similar identifier column in the on-demand data table, the on-demand data table can be added and configured using the following steps:

## Procedure

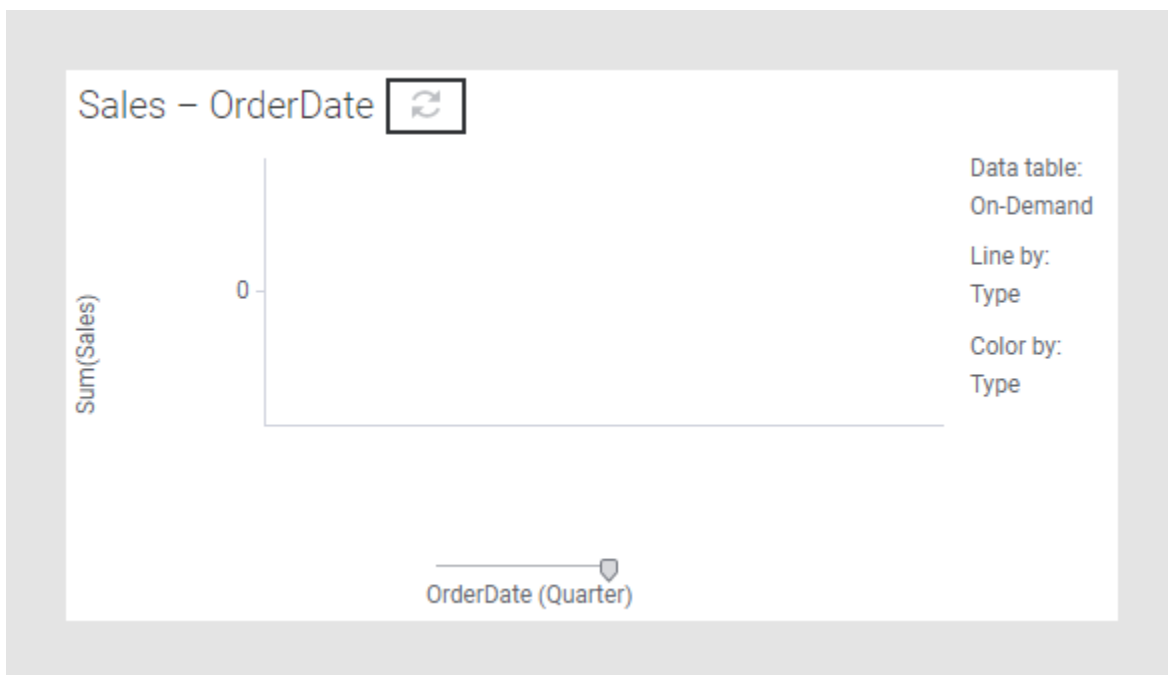
1. On the [authoring bar](#), click **Files and data** .
2. In the **Files and data** flyout, locate the data source (a data connection or an information link ) of interest.
3. When you have selected your data, the **Add data to analysis** summary view is shown. Make sure that the new data is added as a new data table, if this is what you want.
4. From the drop-down list, showing **Import** or **External**, select **On-demand**.
5. Click **Configure**.
6. In the On-Demand Settings dialog, under **Define input for parameters that should control loading**, click to select the column in the new data that contains the identifiers.
7. Click **Define Input**.
8. In the Define Input dialog, under **Input for the selected parameter**, select **Values from column**.
9. Select the **Data table** used by the main visualization that you want to use to control the details visualization and the on-demand data.
10. Select the **Column** containing identifiers in the main data table.
11. Make sure that the **Marked rows** check box is selected, and that only the check box for the marking used in the main visualization is selected.
12. Click **OK** to close the Define Input dialog.
13. In the examples below, the **Load automatically** check box to has been cleared to use manual update only.
14. Click **OK** to close the On-Demand Settings dialog.
15. Click **OK** to add the on-demand data.  
The on-demand data table is loaded. The data shown in any visualization based on the on-demand data table will depend on what is marked in the main visualization.

## Result

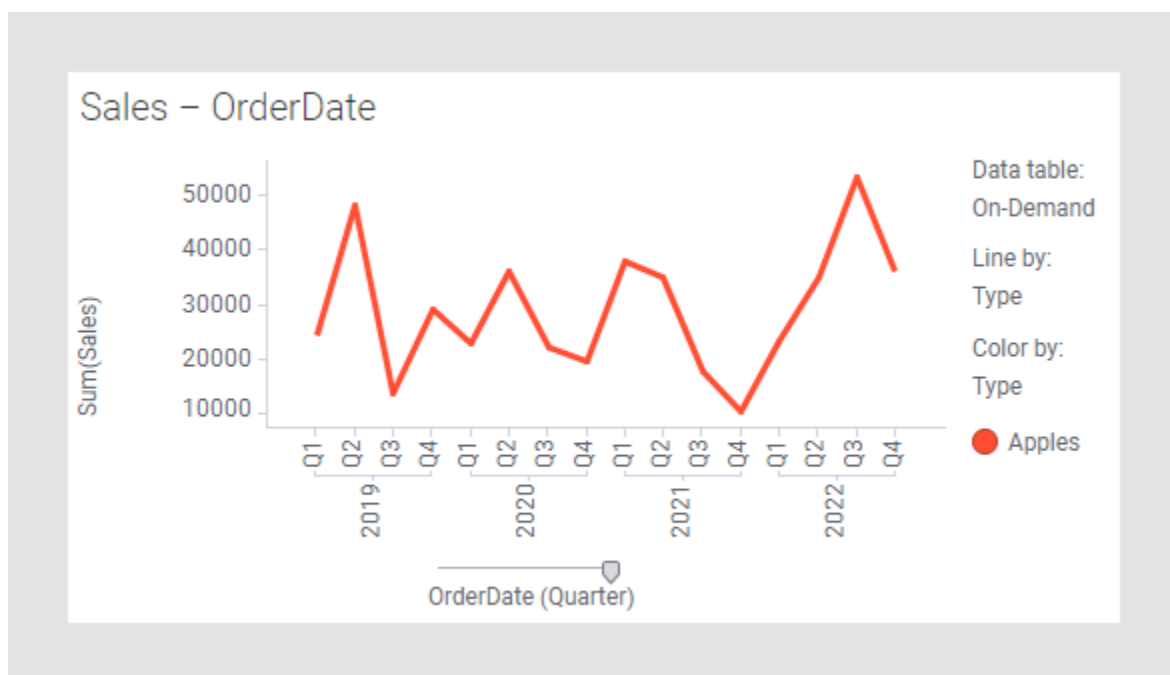
Assume that a manual update was selected (the **Load automatically** check box was cleared), and you have created a main visualization that looks as follows:



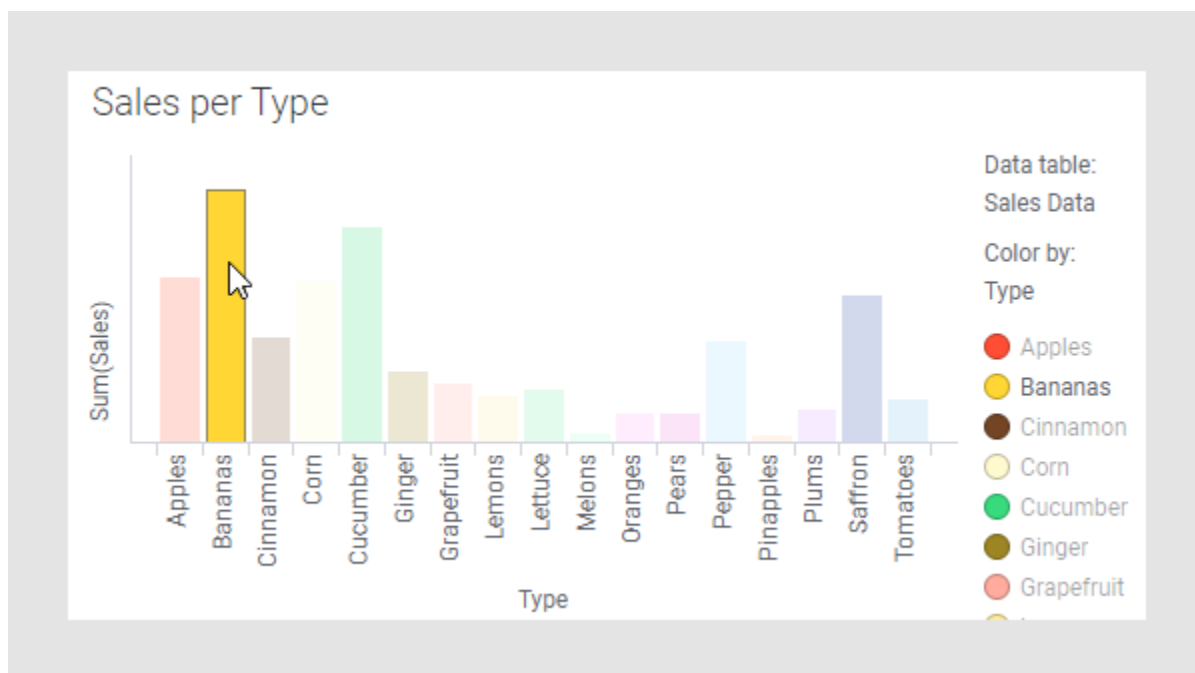
Another visualization is added, based on the on-demand data table. When clicking an item in the main visualization, the refresh button is shown on the title bar of the visualization based on the on-demand data. (If nothing was marked from the beginning, the on-demand visualization will be empty until the first refresh.) Click **Refresh data table**.



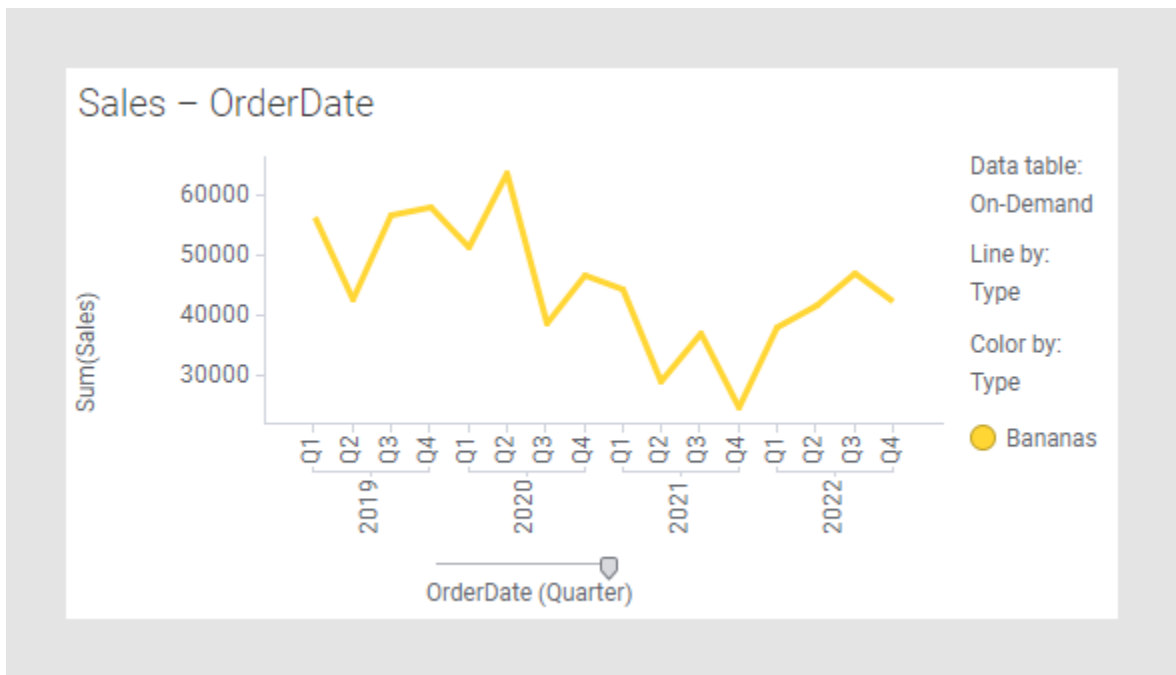
The visualization is updated to show details about the marked item:



Clicking a different item in the main visualization once again shows the refresh button in the on-demand visualization.



Click refresh to update the on-demand visualization to use the new marking:



### ***An on-demand data table controlled by a property***

The data that is to be loaded on demand can be controlled in a number of ways. The example below uses a property control in a text area to select which data to show in a bar chart based on an on-demand data table.

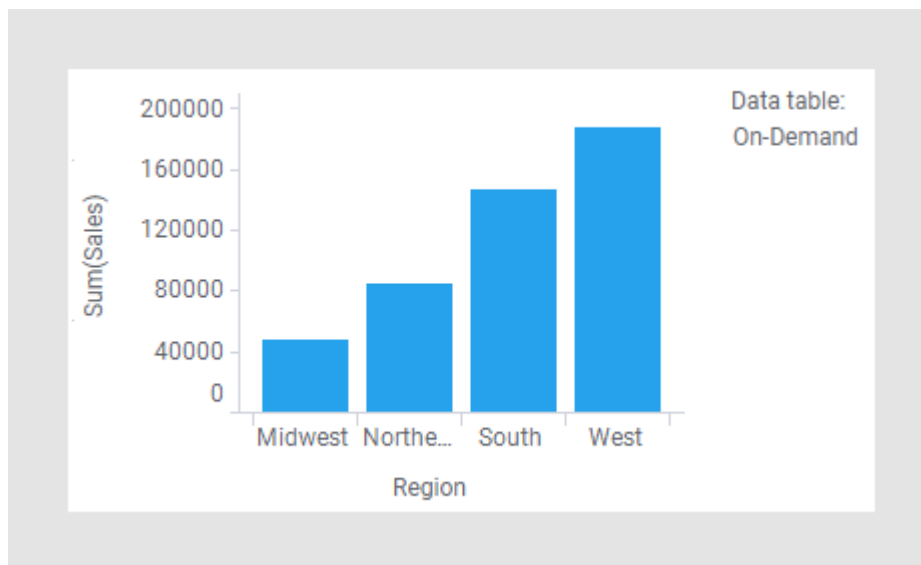
In this example, we first assume that we have a data table containing a string column called "Type" which lists a number of different product types loaded in the analysis. We also assume that there is a data connection with some additional data available, which also contains a "Type" column. See *Adding Data Connections to an Analysis* or *How to Work with Data Connections and their Data Sources* in the Library in the *Spotfire Analyst User Guide* for more information about connections.

#### **Procedure**

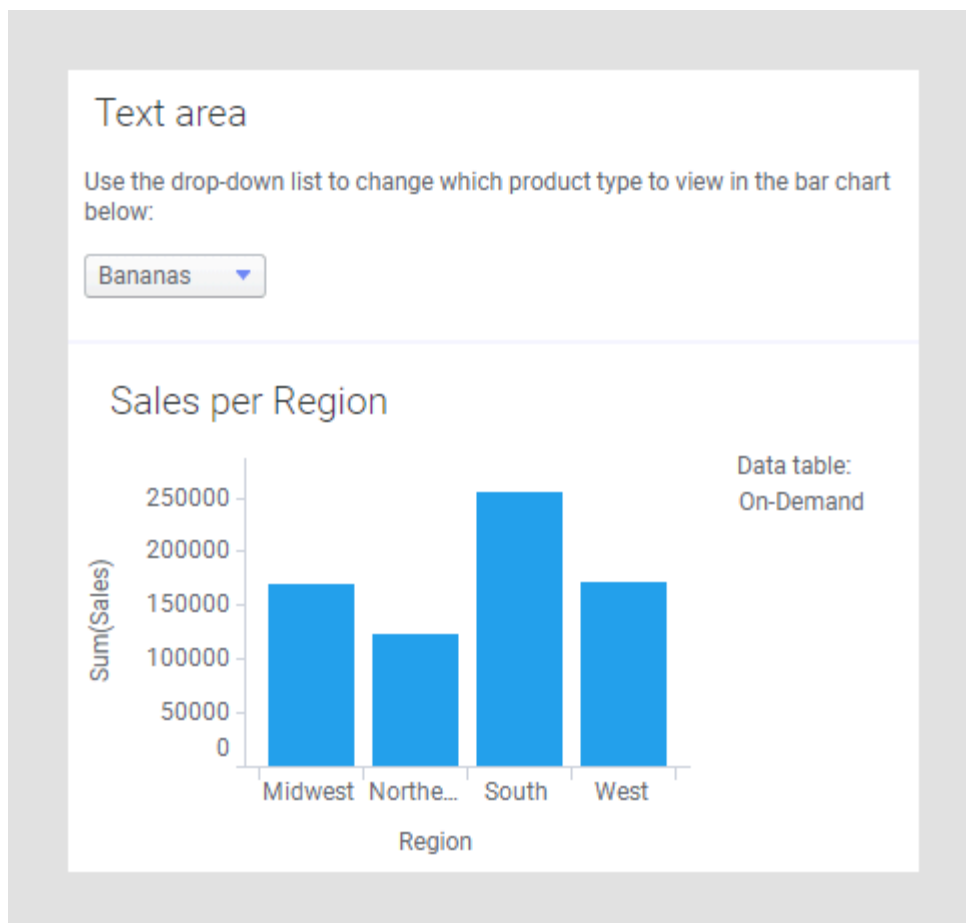
1. Follow steps 1-5 in the example above to open the On-Demand Settings dialog.
2. In the **Define input for parameters that should control loading** list, click to select the column containing the product types ("Type").
3. Click **Define Input**.
4. In the **Input for the selected parameter** list, select **Values (fixed/properties/expression)**.
5. Click the **Property** radio button.
6. Click **Select**.
7. In the Select Property dialog, if no suitable property is available, click **New** (in the Document Properties tab).
8. In the New Property dialog, define a string property using one of the available product types as default value. For example, create a string property called "Type" with the value "Apples".

9. Click **OK** in all dialogs and the flyout.

The on-demand data table is loaded using the limiting default value and can be used in visualizations. In the example below, the visualization used is a bar chart showing the sum of sales for Apples in four different regions.



By adding a property control to a text area, other people can easily change the value of the document property and, hence, update the on-demand visualization. See [Adding a property control to a text area](#) on page 325 for more information.



If **Load automatically** was selected in the on-demand settings, the visualization will be updated each time the property is changed via the control. If **Load automatically** was cleared, a refresh button will be displayed on the title bar of the visualization each time the input is changed.

## Conditions to control on-demand data

When working in the installed client, you can specify to load data from data connections or from information links on demand, based on different conditions.



On-demand data must be authored in the installed client.

The on-demand loading can be controlled by specifying one or more conditions that must be met for data to be loaded. See [Configuring an on-demand data table](#) on page 61 for some examples.

Conditions can be set in a number of different ways. For example, they could be determined by the value of a property or an expression, or by the values of the filtered or marked rows in a column from another data table. If a selected data connection or information link has been configured with required prompts or parameters, then these will automatically be required parameters for the on-demand loading and you must specify a condition using the **Define Input** button for each required parameter.

If desired, you can apply a transformation to the data prior to loading. While the transformation will be performed on the data corresponding to the condition only, you might gain some performance by doing the transformation here rather than conditioning an entire information link directly.

## Examples of conditions

What to control on-demand loading	How to set it up
Marking in another data table	<ol style="list-style-type: none"> <li>1. Click to select the column of interest from the <b>Define input for parameters that should control loading</b> list.</li> <li>2. In the Define Input dialog, set Input for the selected parameter to <b>Values from column</b>.</li> <li>3. Select the <b>Data table</b> from the analysis where you want to mark data.</li> <li>4. Select the <b>Column</b> from the selected data table to match against the column in the on-demand data table.</li> <li>5. Select to Limit by <b>Marked rows</b> by selecting the check box for the desired marking.</li> </ol>
Filtering in another data table	<ol style="list-style-type: none"> <li>1. Click to select the column of interest from the <b>Define input for parameters that should control loading</b> list.</li> <li>2. In the Define Input dialog, set Input for the selected parameter to <b>Values from column</b>.</li> <li>3. Select the <b>Data table</b> from the analysis where you want to mark data.</li> <li>4. Select the <b>Column</b> from the selected data table to match against the column in the on-demand data table.</li> <li>5. Select to Limit by <b>Filtered rows</b> by selecting the desired filtering scheme from the drop-down list.</li> </ol>



What to control on-demand loading	How to set it up
A range of values defined by the min and max values from the current marking or filtering, for a selected column	<ol style="list-style-type: none"> <li>1. Click to select the column of interest from the <b>Define input for parameters that should control loading</b> list.</li> <li>2. In the Define Input dialog, set Input for the selected parameter to <b>Range from column</b>.</li> <li>3. Select the <b>Data table</b> from the analysis where you want to mark data.</li> <li>4. Select the <b>Column</b> from the selected data table to match against the column in the on-demand data table.</li> <li>5. Select to Limit by <b>Marked rows</b> or <b>Filtered rows</b> by selecting the corresponding marking or filtering scheme.</li> </ol>
A document property value	<ol style="list-style-type: none"> <li>1. Click to select the column/parameter of interest from the <b>Define input for parameters that should control loading</b> list.</li> <li>2. In the Define Input dialog, set Input for the selected parameter to <b>Values (fixed/properties/expression)</b>.</li> <li>3. Click the <b>Property</b> radio button.</li> <li>4. Click <b>Select</b> and specify which document property to use in the dialog that opens.</li> </ol>
An expression	<ol style="list-style-type: none"> <li>1. Click to select the column/parameter of interest from the <b>Define input for parameters that should control loading</b> list.</li> <li>2. In the Define Input dialog, set Input for the selected parameter to <b>Values (fixed/properties/expression)</b>.</li> <li>3. Click the <b>Expression</b> radio button.</li> <li>4. Click <b>Edit</b> and specify your custom expression.</li> </ol>
All values over (or under) a certain limit, for example: Sales > 1000	<ol style="list-style-type: none"> <li>1. Click to select the column/parameter of interest from the <b>Define input for parameters that should control loading</b> list.</li> <li>2. In the Define Input dialog, set Input for the selected parameter to <b>Range (fixed/properties/expression)</b>.</li> <li>3. In the field of interest (e.g., Min) click the <b>Fixed value</b> radio button.</li> <li>4. Type the value of interest in the field or click <b>Select</b> to pick a value from the available values in the column.</li> </ol>

The data retrieved for the on-demand data table can be based on a combination of all of the examples above.



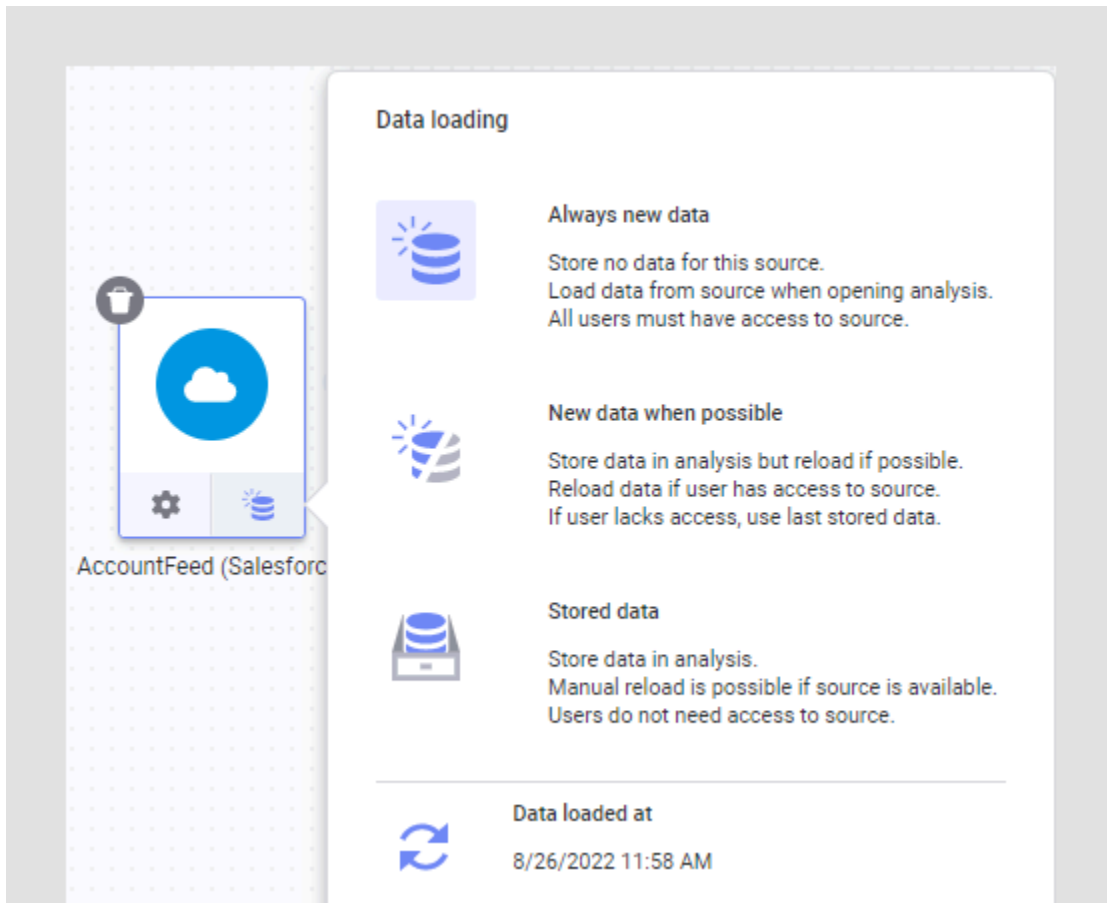
Special attention is needed when setting up an on-demand data table dependent on nothing but a parameter defined within a configuration block (a text file which configures the initial state of an analysis, see [Create a Configuration Block](#) on the [Community](#)). An on-demand data table must always have at least one input defined for the parameter in the **On-Demand Settings** dialog. Because configuration block parameters will automatically be assigned to document properties (if they exist and have the appropriate data type), a document property may act as a bridge between a configuration block parameter and a parameter in an information link. If another input is defined, such as a column filter, no such bridge is required and the parameter from the configuration block will be used automatically.

## Linked, stored, and embedded data

When you are about to save an analysis, you should decide whether data should be stored in the analysis, or whether it should be updated when the source data is updated.

By default, data always remains linked to the source, but data from different sources have different default loading behaviors. Usually, you can change the loading behavior for each particular source in an analysis. For example, this makes it possible to always get the latest data from a Salesforce instance, each time an analysis is opened, whereas target data from a local file can be stored in the analysis and only be updated should the targets change. See [Storing data within the analysis](#) on page 193.

The data loading settings are, in general, available when clicking on a data source node in the [Data canvas](#), but you can also change settings directly when [saving an analysis](#).



See also [Load methods](#) on page 58 and [Reloading data](#) on page 227.

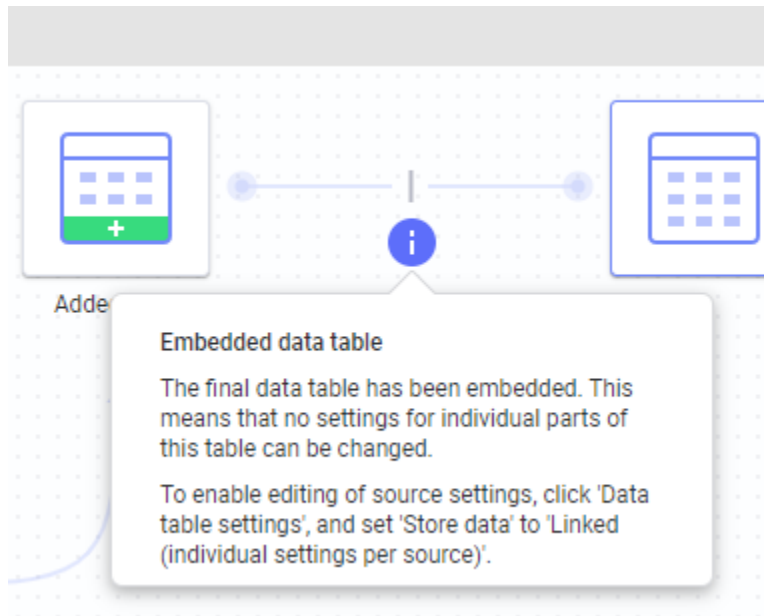
### Embedded data

In the data table Settings popover (in the data canvas), there is an additional setting available for **Stored data** (in-memory only). There, it is possible to switch from **Linked (individual settings per source)** to **Embedded in analysis**. This setting was used more before the data loading settings for linked data were added. However, there might be some cases where the Embedded data setting still is preferred. Which option to choose depends on the data you use and who the end users of the analysis are; and also which data sources they have access to. The linked option provides more flexibility, because it allows you to select different data loading settings for different parts of a data table, as discussed above. Therefore, the linked option is most of the times the preferred solution.

- **Linked** data gives you the option to always retrieve the latest data from the data source, each time the analysis is opened, but you can also choose to store data for a certain part of the data table in the

analysis. As a third option, you can choose to get new data when possible, but to store data for those users that lack access to the original source, or for use when offline. Using linked data allows you to keep the file size to a minimum.

- **Embedded** data captures a snapshot of the final data table in the analysis file, so that the analysis file is self-contained with data and possible to use offline. The snapshot can be updated manually, see [Reloading data](#) on page 227. Embedded data can in some cases save memory, for example, if rows have been removed from the original data. However, if you embed the final data table you lose the ability to configure different data loading for different parts of the data table, and to edit settings and configurations for each node that builds up the data table.



In-database data can never be embedded because it is always fetched directly from the database.

### Calculated columns

Calculated columns can be cached in the analysis file, even when the main data table is linked, as long as the Data loading setting is Stored data. The values in the calculated columns are then based on the values from the most recent reload of the ordinary data columns. When an analysis is opened, ordinary data columns might be reloaded from their data sources (if they are linked). Only those calculated columns where the original data has changed will then be recalculated. The next time the analysis is saved, the new values in the calculated columns are cached in the DXP file. You can switch off caching of calculated columns in the data table Settings popover (in the data canvas), to always recalculate the calculated columns and reduce the file size of the analysis file.

If a calculated column uses a time dependent function, such as `DateTimeNow()`, it will always be recalculated when the file is opened, even for embedded data tables.

### Reloading data for each user when using scheduled updates


If an analysis has been configured to use automated data updates on the server, so called scheduled updates, data is regularly being preloaded with fresh data at a certain time (for example, on a daily basis) to make the loading time faster for the end users. However, an analysis can contain data from several different data sources and only some of them might contain shared data that should be reloaded at specified, scheduled times, whereas others should be reloaded on a per-user level (for example, when using personalized data connections). If some data sources should always be reloaded when a user opens the analysis, even if the analysis has been configured for scheduled updates, this can be specified per data source.

For more information about scheduled updates and how to configure them, see [Scheduled updates to analyses](#) in the *Spotfire® Server and Environment - Installation and Administration* manual.

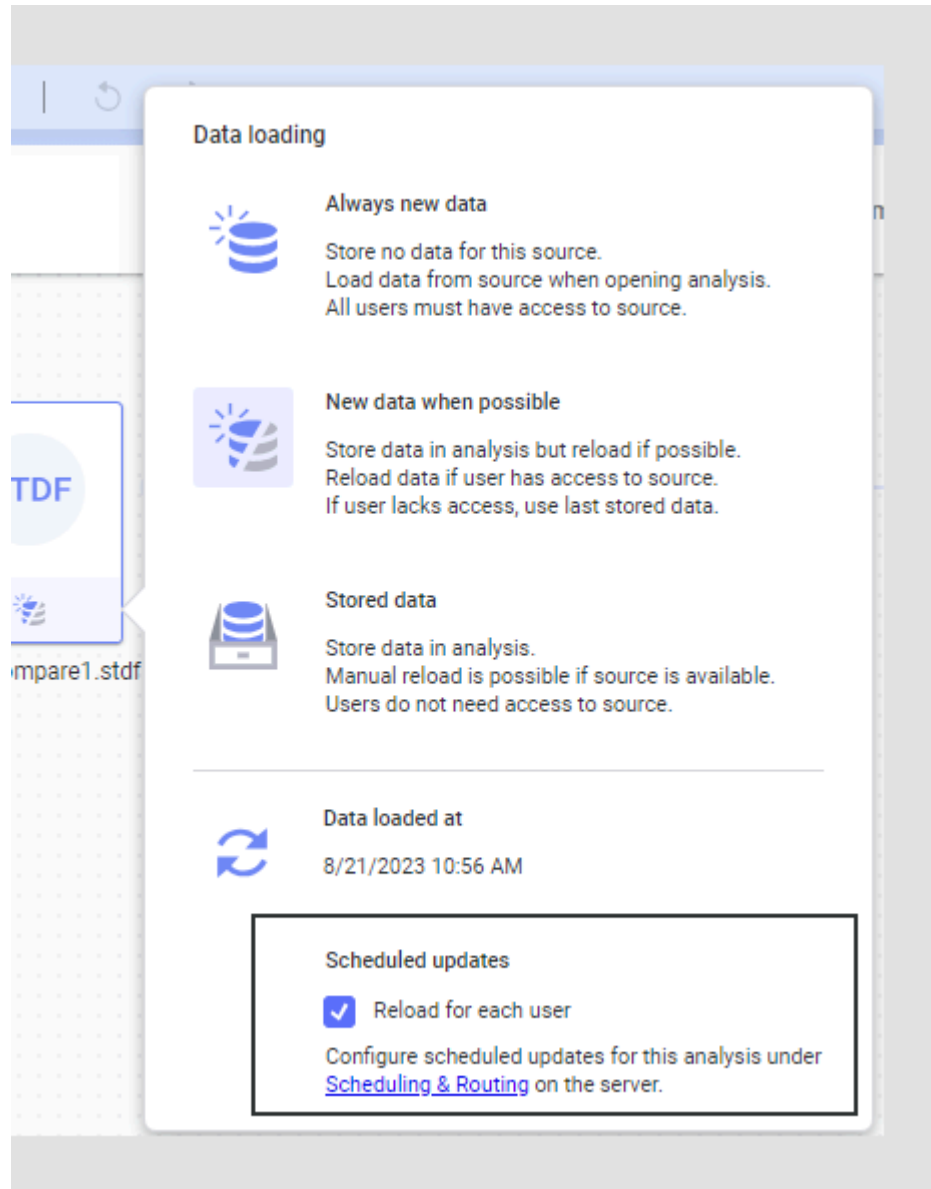
### Prerequisites

- An analysis using linked and imported data has been, or will be, configured with a schedule update using the **Scheduling & Routing** page in the Spotfire Server web administration pages (or using an external process).
- You have an analyst or business author role with access to the data canvas (the **Manage data tables** license feature).

### Procedure

1. Open the analysis in a Spotfire client.
2. On the [authoring bar](#), click **Data canvas** .
3. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.
4. Locate the node that represents the data source of interest and click on the button at the lower part of the node representing the data loading settings.

5. At the bottom of the popover, under **Scheduled updates**, select **Reload for each user**.



The data source will always be reloaded for each new user when the analysis is opened. All other data sources (with the check box cleared) uses the preloaded data from the scheduled update.



This setting will have no effect unless a scheduled update has been configured for the analysis on the server.

## Adding more data

Data can be added to the analysis in several different ways: as new columns, as new rows or as new data tables. All of these methods are available when you add data from the **Files and data** flyout. If Spotfire identifies that new data seems to fit in an already available data table, you might get a recommendation to add the data as new rows. However, you can always ignore the recommendation.

See also [Choosing how to load the data](#) on page 55.

Adding data as separate data tables is useful if the new data is unrelated to the previously opened data table or if the new data is in a different format (pivoted vs. unpivoted). Data can be added from local files on your computer, or from different types of data files that have been previously saved in the

library using the Windows client. You might also have access to other data sources from where you can add data.


Data from different data tables can be viewed in the same visualization, provided that there is at least one column in both data tables that can be used to match the data from the two data tables. You can, of course, also create separate visualizations for different data tables.

In some cases, when you need to bring in-memory data from different data sources together in a single visualization, it may be more suitable to add columns or add rows to a data table, rather than keeping two different data tables.



You can also drag a supported file to the analysis directly and drop it in your browser.

## Procedure

1. On the [authoring bar](#), click **Files and data** .
2. In the flyout, locate the data of interest by searching for a suitable filename or keyword, or by browsing the different categories.

You can search for specific types of data by typing keywords. For example, enter `type:dxp` (to search for analyses), `type:dataconnection` (to search for data connections), or, `type:query` (to search for information links). See [Searching the library](#) for more details.

3. Select the source data and specify any required settings.



If the data you are about to add matches other in-memory data in the analysis, then you will be recommended to add the new data as rows in a matching data table, instead of as a new data table. It is often easier to visualize the data if you have a single data table, so use the recommended solution whenever possible. See also [Adding rows to a data table](#) on page 182.



When a data connection is added to the analysis, all views in the connection will be available within the analysis until the connection or a view is removed, regardless of whether the view is added as a data table or not. Therefore, you should select a connection under **Data connections in current analysis** (in the **Files and data** flyout, under **Recommended**) to add more views from an already used connection rather than adding another connection to the same source.

4. When you have selected your data, you will see a summary view of all data tables that will be added or updated when you click OK. You may have the option to change how data is loaded, depending on what source data you select: If the selected source is a data connection you can change the load method of the data table. See [Choosing how to load the data](#) on page 55 for more information.
5. If you want to add more data tables, repeat for each data table.

## What to do next

If the data was added as new data tables, determine whether the new data tables should be related to each other or to previously added data tables. Remember that you must define a relation if the new data table is to be used to create [details visualizations](#) for the previously added data tables. Read more in [Related data tables, joins and column matches](#) on page 74.

## Related data tables, joins and column matches

Depending on how you access your data and which client you use, there are several different ways to link data from different sources together using Spotfire. With in-memory data, you can add new data as columns or rows to an existing data table. If you use data from data connections, you might be able to create a structural relation already on the database side, when creating the data connection. If you choose to keep your data as separate data tables in Spotfire, you can either create a relation between

the data tables to support brush-linking, or, simply use columns from different data tables directly in a visualization, if they have at least one column match.

The different ways of bringing data together into Spotfire provide various degrees of tightly coupled data.

### Add columns or rows to a data table

In some cases, when you need to bring in-memory data from different data sources together in a single visualization, it might be more suitable to add columns or add rows to a data table, rather than keeping two different data tables.

If Spotfire identifies that new in-memory data seems to fit in an already available data table, you might get a recommendation to add the data as new rows. See [Adding more data](#) on page 73 for more information. However, you can always ignore the recommendation. You can also choose to add new data, as new rows or as new columns, to a specific location in the data table structure from the [data canvas](#).

When you add data as new columns, you can choose which type of join to use in the Add columns – match columns dialog, and use the interactive example in the dialog to see the difference between the join types.

### Structural relations between database tables in connections

With in-database or in-memory data tables accessed using a data connection, you can often join several database tables into a single virtual data table before adding it to Spotfire using key columns (primary/foreign keys).

When you set up data connections from relational or other non-cube data sources, you have the possibility to use relations created by the database administrator or add new relations between original database tables in one data connection, so that they in Spotfire are joined to a single view (or data table). See [Details on Views in Connection](#) in the Spotfire Analyst help for more information.

### Joins in information links

If data is fetched through information links (can be defined using the installed client only), it is possible to define joins between different database tables using [Information Designer](#).

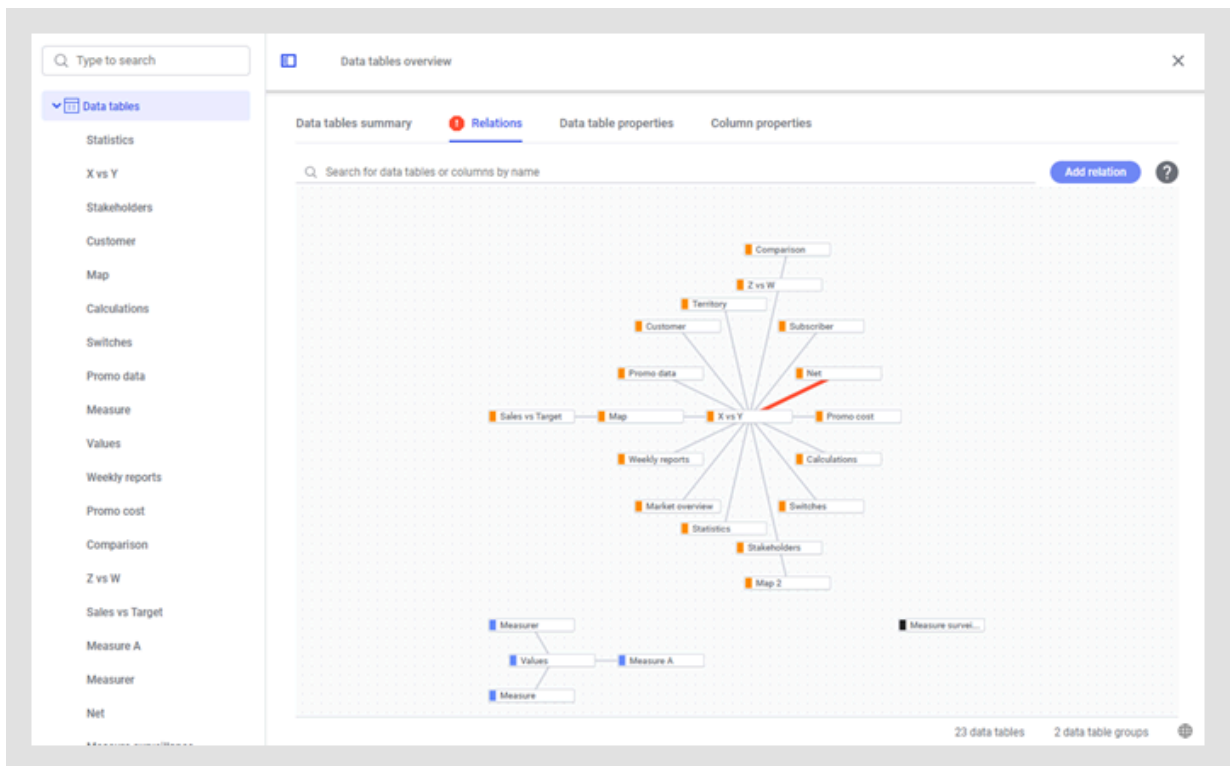
### Relations between data tables

You can create relations between different data tables in Spotfire without actually joining them. This will form a looser connection between the tables but it can be used if you want to set up a details visualization using one of the data tables, to be limited by selections in the other, or just to provide brush-linking, meaning that you can mark or filter items in a visualization based on one data table and automatically get the same items marked or filtered in a visualization based on the related data table. When data tables are related, marking is always carried over, but filtering can be configured in different ways. See [Filtering in related data tables](#) on page 590 for more information.

For a relation to be useful, you must have one or more key columns (identifier columns) available in both data tables, and use these to define which rows in the first data table will correspond to rows in the second data table. The matching can also be configured using expressions that combine multiple columns, or modify the values in the columns. If you need more than one key column to define a unique identifier, you can add one column pair for each identifier column.

Relations are defined in the **Data canvas**. On the **Relations** tab in the [Data tables overview](#), you can easily see if there is anything wrong with a relation and you will quickly find groups of data tables where relations have been specified. See also [Adding or editing a relation between two data tables](#) on page 202.





You can also add some simple relations using the **Link data tables** recommendation, which can show up if you select a suitable ID column for one of your data tables in the **Data in analysis** flyout.

## Column matches

You can often combine data from different data tables in one visualization without having a relation between them, if at least one column is available in both data tables. If two columns are of the same data type and have the same name, Spotfire will match them automatically during loading.

Column matches is the only option to use when you want to visualize data from two separate in-database sources, which cannot be joined in other ways. However, you can also use column matches, for example, if you have a too large data volume to be able to join sources directly, but when visualizing the data you only use aggregated values for the columns you match on.

In the installed client, you can view, create and edit column matches in the Data table properties dialog. Click **Data > Data table properties** and go to the **Column matches** tab to define column matches.

## Using a different data table to show a line

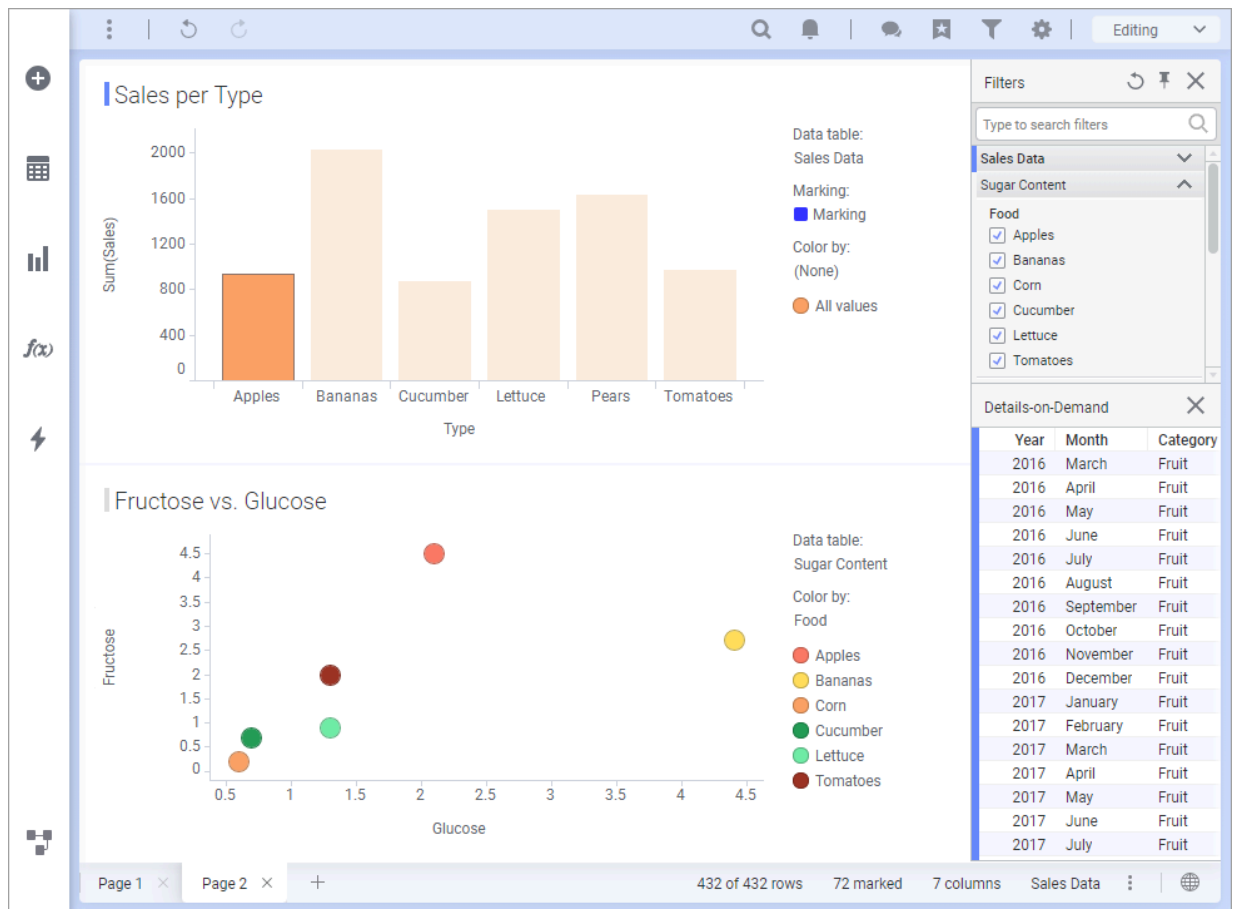
You can show a simple line from a different data table in a scatter plot. See [Details on Line from Data Table](#) in the Spotfire Analyst help.

## Independent data tables

This is an example of independent data tables. These two visualizations are placed on the same page, but they are not related to each other. The visualizations correspond to separate data tables. Marking or filtering in one visualization will not affect the other when they are independent. The Details-on-Demand shows information about the marked item in the active visualization.

In this example, the bar chart shows the sum of sales for different types of fruits and vegetables. The scatter plot shows the content of fructose and glucose for different types of fruits and vegetables.





## Related data tables

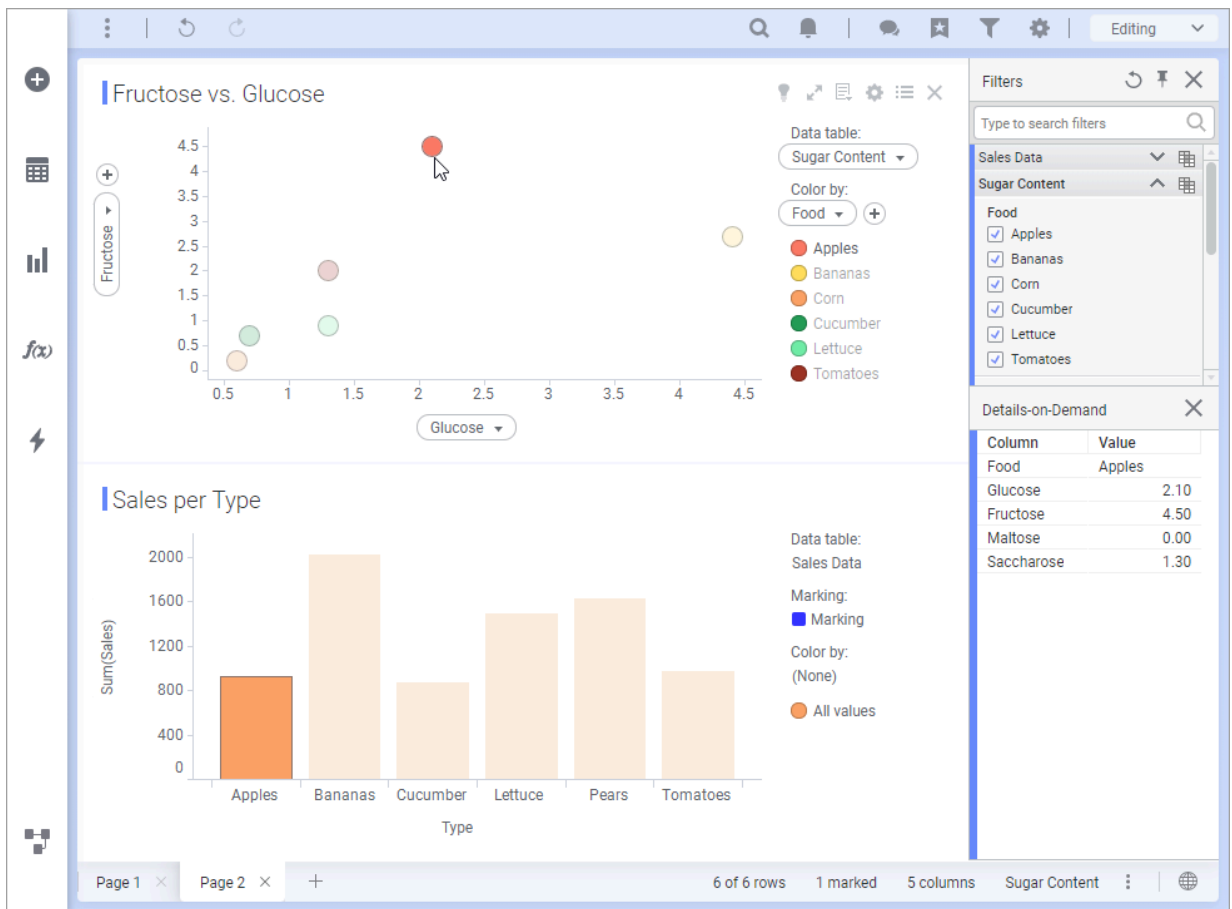
This is an example of related data tables. The visualizations are based on different data tables that are related (the data tables have been linked together by defining a relation between them). Marking items in one visualization will mark the corresponding items in the related visualizations. Filtering data in one data table can filter the related data in the other data tables.

See [Adding or editing a relation between two data tables](#) on page 202 for information on how to define relations.



Related visualizations can be placed on different pages in an analysis. This means that markings that are not visible at the moment can affect the visualizations that you are looking at.

In this case, two data tables with information about fruit and vegetables are related. The scatter plot shows the amount of glucose and fructose for different types of fruits and vegetables, while the bar chart shows the sum of sales for the same types of fruits and vegetables. Marking an item in the scatter plot, in this case the one with the highest level of fructose (Apples), will mark the Sum(Sales) for Apples in the bar chart.



## Main and details visualizations

This is an example of multi-step main and details visualizations. The visualizations in this example are based on the same data table and show different levels of detail. However, the visualizations could just as well be based on data from different data tables. Marking in one visualization defines the data of the next visualization, making it possible to drill down in level of detail.

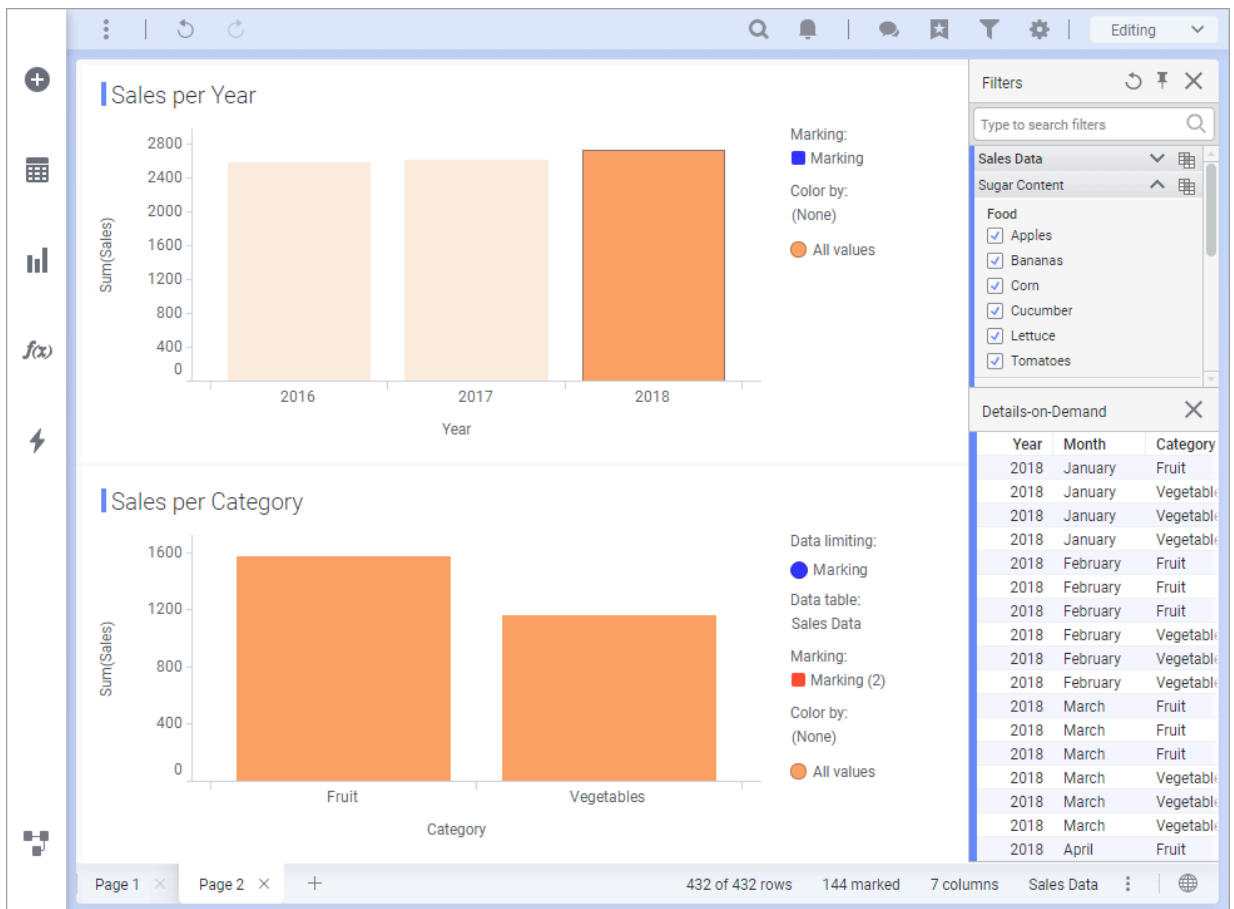


Related visualizations can be placed on different pages in a visualization. This means that markings in a visualization that is not visible at the moment can affect the visualization that you are looking at. If a visualization is empty, it might be because it is based on markings from another visualization. Go to the main visualization and mark an item to show information in the details visualization. When creating the analysis, you can add a message explaining in which visualizations to mark items to see the details. See [Drilling down into details](#) on page 597 for more information.

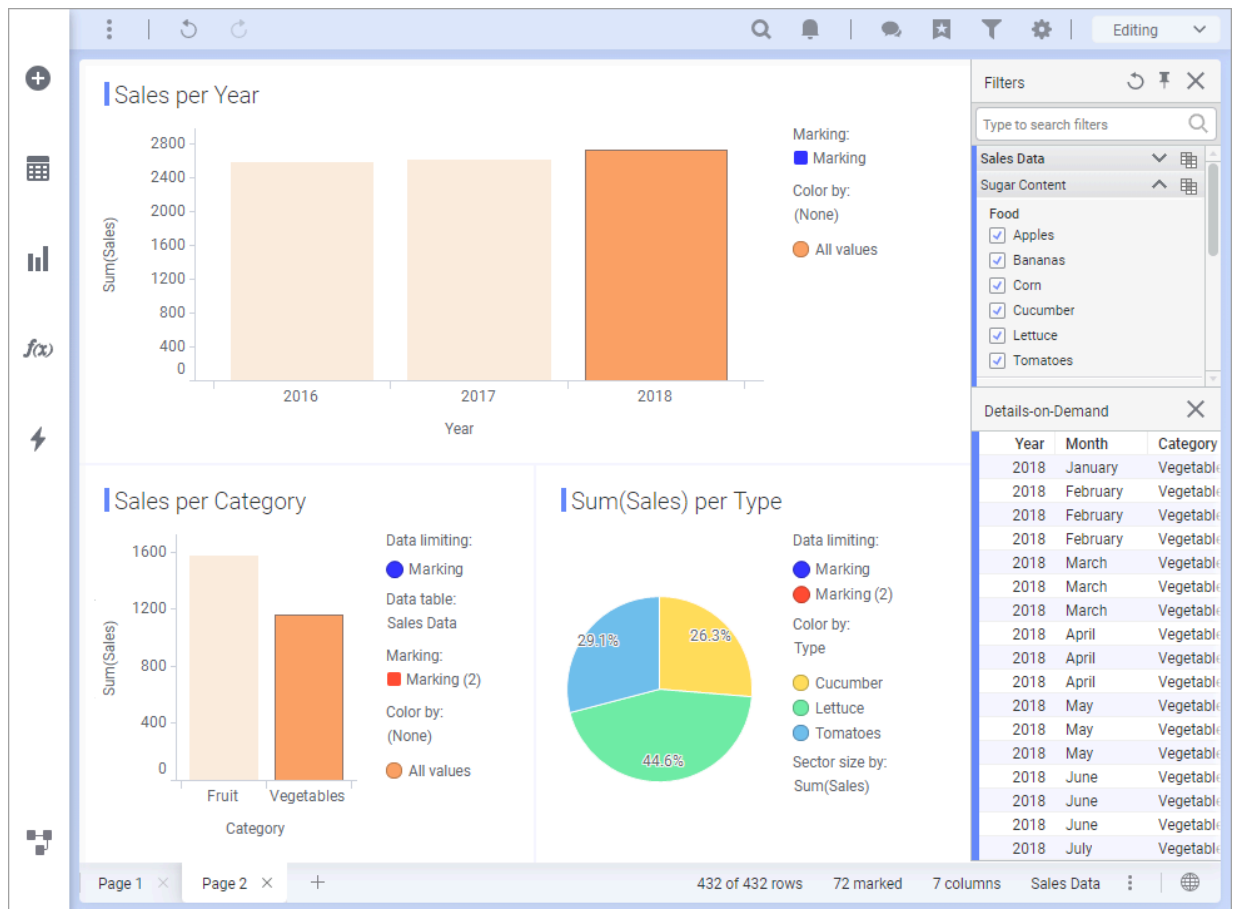


The Details-on-Demand shows information about the marked rows from the active visualization; it could be either the main or the details visualization.

In this example the main visualization shows Sales per Year. If you mark a year, for example, 2018, in the main visualization, data from that year will be shown in the details visualization. This details visualization shows Sales per Category (fruit and vegetables) for 2018.



The marking of an item in the next visualization, Sales per Category, can also be configured to show an even more detailed visualization. Below, another visualization has been created where marking the category "Vegetables" in Sales per Category shows the percentage of sales per type (cucumber, lettuce and tomato) in that category for 2018.



This image shows three different visualizations presenting different aspects of data from the same data table.

## Added columns

By adding columns or rows, it is possible to combine data from different sources into a single data table that can be used in a visualization.

In this example, a data table containing information about the cost and sales for different kinds of fruits and vegetables (Table 1) has been joined together with another data table containing information about the content of Glucose, Fructose, Maltose and Saccharose (Table 2) per fruit and vegetables. In this case, two columns from Table 2 have been added (Glucose and Fructose) to Table 3.

Table 1

Food	Sales	Cost
Apples	12	10
Pears	21	13
Bananas	29	26
Cucumber	9	6
Tomatoes	13	11
Lettuce	22	20

Table 2

Food	Glucose	Fructose	Maltose	Saccharose
Apples	2.10	4.50	0.00	1.30
Bananas	4.40	2.70	0.00	6.40
Corn	0.60	0.20	0.30	2.30
Cucumber	0.70	0.70	0.00	0.00
Lettuce	1.30	0.90	0.00	0.00
Tomatoes	1.30	2.00	0.00	0.00

Table 3


Food	Sales	Cost	Glucose	Fructose
Apples	12	10	2.10	4.50
Pears	21	13		
Bananas	29	26	4.40	2.70
Cucumber	9	6	0.70	0.70
Tomatoes	13	11	1.30	2.00
Lettuce	22	20	1.30	0.90

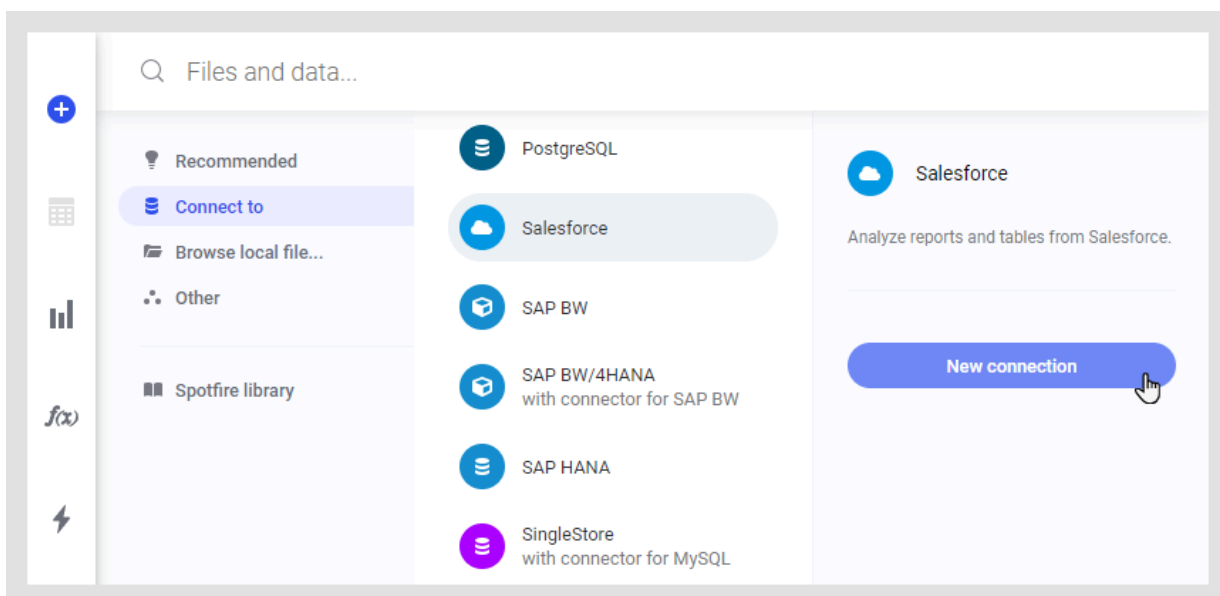
See [Adding columns to a data table](#) on page 187 and [Adding rows to a data table](#) on page 182 for more information.

## Data from databases

In Spotfire, you can connect to and access data from your external data sources, such as databases, data lakes and warehouses.

### Accessing data from supported data sources with connectors

Spotfire has native support for lots of data source types. You can connect to all supported data sources from the **Connect to** list in the **Files and data**  flyout.



Technically, when you connect to a data source from the **Connect to** list, you're using a connector. Each connector is designed and optimized for one or more data sources. For the full list of supported data sources and the connectors you use to access data from them, see: [Supported data sources](#) on page 86.

If your data source is available in the **Connect to** list, this is the preferred way to access your data. Connectors are well-suited for handling big data, because data can be kept external for in-database analysis. That means that queries are pushed to the data source from Spotfire as you work on the analysis. Of course, with connectors you can also import data into Spotfire or load the data on demand.

To get started and create your first connection to your data source, see: [Accessing data from a database](#) on page 82.

## Connecting to other data sources

Even if Spotfire does not yet have native support for the type of external data source where you store your data, meaning there is no specifically designed and optimized data connector for that source, you can probably still load your data into Spotfire.

To connect to a data source for which there is no self-service data connector, you can use one of Spotfire's generic methods for connecting to external data sources.

### Information Designer - Server-side JDBC connectivity

With [Information Designer](#) (Information Services), you can connect to and load data from JDBC-compliant data sources. Data can be imported for in-memory analysis in Spotfire.

Information Services is a server-side method for loading data, which means that JDBC drivers and JDBC data source templates that are required to connect to your data source must be installed and configured on your Spotfire Server.

Once these preparations are set up on the Spotfire Server, you use the Information Designer tool in your Spotfire Analyst client to create information links for loading data from your JDBC-compliant external data source.



Because Information Services runs on Spotfire Server, this method for loading data is not available if you are running TIBCO™ Cloud® Spotfire or Spotfire® Analyst without a server.

### Load data with ODBC, OLE DB, or ADO.NET data provider

If there is no native data connector that you can use to connect to your external data source, and you cannot use Information Services to connect with generic JDBC connectivity, you can use Spotfire's generic support for loading data with a data source driver that you have installed on your computer.

With this method, you can use a driver of the types ODBC, OLE DB data provider, and ADO.NET data provider to connect and load data from an external data source.



The [Load data with ODBC, OLE DB, or ADO.NET data provider](#) feature provides only limited performance and functionality for loading data. Therefore, only use it if you are not able to use a connector or Information Services to connect to your data source.

## Accessing data from a database

You can connect to and access data from external data sources, such as databases, data warehouses, and data lakes. On this page, you learn how to connect to your data source with a data connection.

### Checking that a data source is supported

First check the list of supported data sources to see that your data source is supported. Also visit the system requirements for details about supported versions.

- [Supported data sources](#) on page 86
- [System requirements](#)



If your data source is not in the list, it means that Spotfire does not have native support for it with a data connector. Instead, you can try to connect to your data with Information Services.



When you use Spotfire in a web browser, you can only create and edit data connections to some data source types. To learn more, see [Data connections in the Spotfire web client](#) on page 113.

### Installing the required drivers

To access data from many external data sources, a driver is required. If you use a Spotfire installed client, you must install the required driver on your computer.



If you are using Spotfire in your web browser, a Spotfire administrator must install the driver on the computer that is running the Spotfire Web Player service.


1. Go to [Drivers and data sources in Spotfire®](#) and browse to find the data source type that you want to connect to.
2. Find the driver for your data source and install it on your computer.
3. Restart Spotfire.

### Creating a data connection

You can think of accessing data from an external data source as a 3-step process: First you connect to your data source. Then you select the data that you want to access. And finally you decide how you want to load that data in your analysis.

### Connecting to your data source

To create a new connection to your data source, select **Connect to** in the **Files and data** flyout.

1. Open the **Files and data**  flyout, and click **Connect to**.
2. In the list, find and select your data source.
3. In the panel on the right, click **New connection**.
4. In the connection form, enter the information for connecting to your data source.



This step is different depending on the data source type. Usually you must enter a server address, select an authentication method, and log in with your user credentials. Depending on your data source type, there can be many more options. See the documentation for the corresponding data connector for details.

5. To connect to your data source and proceed to selecting your data, click **OK** or **Connect**.

#### Learn more

To learn more about how to configure a connection to a data source, see the following pages:

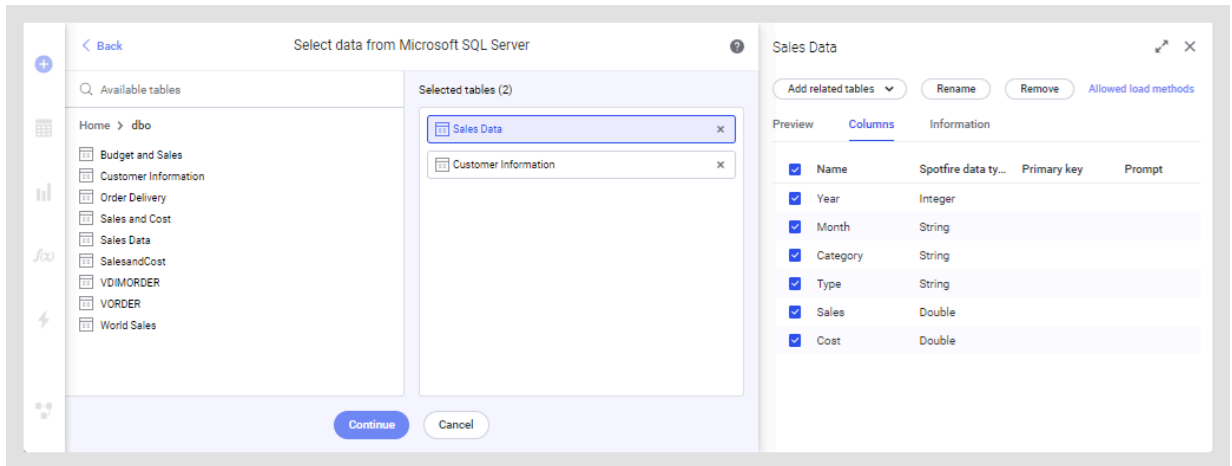
- [Properties in connection data sources](#) on page 115

### Selecting data from your data source

When you have connected to your data source, the next step is to select the data that you want to use in Spotfire.



The user interface for selecting data is different depending on the type of data source, and if you are using Spotfire as an installed or web client. There are limitations to what you can do when using the web client, which you can read about more here: [Limitations when you author data connections in the web client](#) on page 113.



1. In the panel on the left, browse the database tables available in your data source.
2. Click to select the database tables you want to use, and make available as data tables to analyze in Spotfire.
3. For each table, you can decide which columns you want to include.
4. When you are happy with your data selection, click **Continue** or **OK**. In the next and final step, you can choose how you want Spotfire to retrieve the data in your analysis.

### Learn more

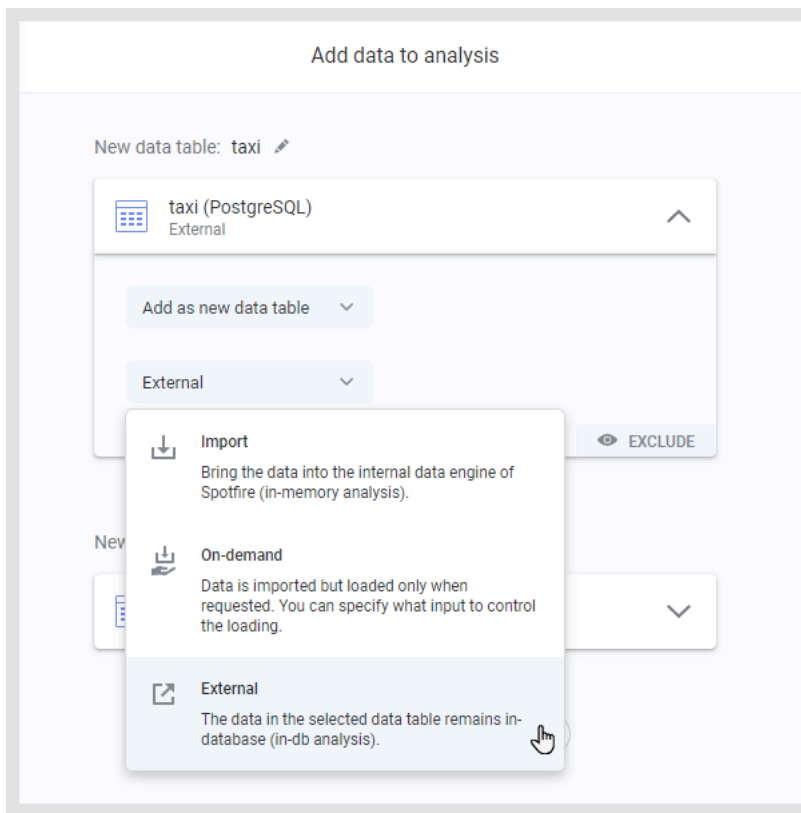
You can also create a more specific or flexible data selection in this dialog. There is a range of tools available, such as:

- **Custom queries** - This is where you can put your SQL skills to use. With custom queries, you can select the data you want to analyze by typing a custom SQL query. You can add and configure custom queries in an installed Spotfire client.
- **Prompting** - Leave the data selection to the user of your analysis file. You configure prompts based on columns of your choice. Then, the end user who opens the analysis can select to limit and view data for relevant values only. For example, she can select data within a certain span of time or for a specific geographic region. You can add prompts to data connections in an installed Spotfire client.

## Selecting how to load the data

When you have selected your data, the final step is to select how Spotfire should retrieve that data, and how to add it to the analysis. In the **Add data to analysis** summary, you can see all the data tables you selected.





1. To see all the options for a data table, click the data table's card. In the expanded card, you can change the load method, and select if the table should be added as a new data table or as rows to a table you already have in your analysis.
2. Click **OK** to finish and add the data to your analysis.

### Learn more

To learn more about different ways to load your data, see the following pages:

- [Load methods](#)
- [Related data tables, joins and column matches](#) on page 74

## Accessing data with a shared connection or information link

You can use data from your databases in Spotfire without having to create a new connection. Data connections and information links can be saved and shared in the library, and you can open them to easily connect to and access a selection of data from an external data source.




In the Spotfire web client, you can open and add data from data connections and information links, even if you do not have authoring capabilities for connections to that data source. See [Data connections in the Spotfire web client](#) on page 113.

### Prerequisites

- The required drivers must be installed on your computer or the computer running the Spotfire Web Player.
- Data source credentials might be required to log in to the data source.

## Procedure

1. On the authoring bar, click **Files and data** .
2. In the Files and data flyout, click **Browse the library**, or use the search bar to locate the data connection or information link that you want to open.




You can use the search bar to display a list of all data connections and information links that are shared in the library. Type the following phrase in the search bar: `type:query` or `type:dataconnection`

3. Click to open the data connection or information link of interest.
  - If required, enter your credentials to log in to the external data source.
4. In the Add data to analysis flyout, you can view and change the settings for how the data will be added. Click **OK** to confirm and add the data to your analysis.

### What is a data connection, and what is an information link?

When you add data to your analysis from a data connection or an information link, it works much the same way. Both enable you to access a prepared selection of data from an external data source. But there are some differences.


#### Data connections

In the library, you can recognize a data connection from this icon .

Data connections are the most convenient way to access data from external data sources in Spotfire. Spotfire has native support for creating connections to many types of data sources. For information on all supported data sources that you can access data from with connections, see [Supported data sources](#) on page 86

When you access data with a data connection, you can choose the load method. This means that you can keep the data external, for in-database analysis, or you can import it, for in-memory analysis. For more information, see [Load methods](#).

#### Information links

In the library, you can recognize an information link from this icon .

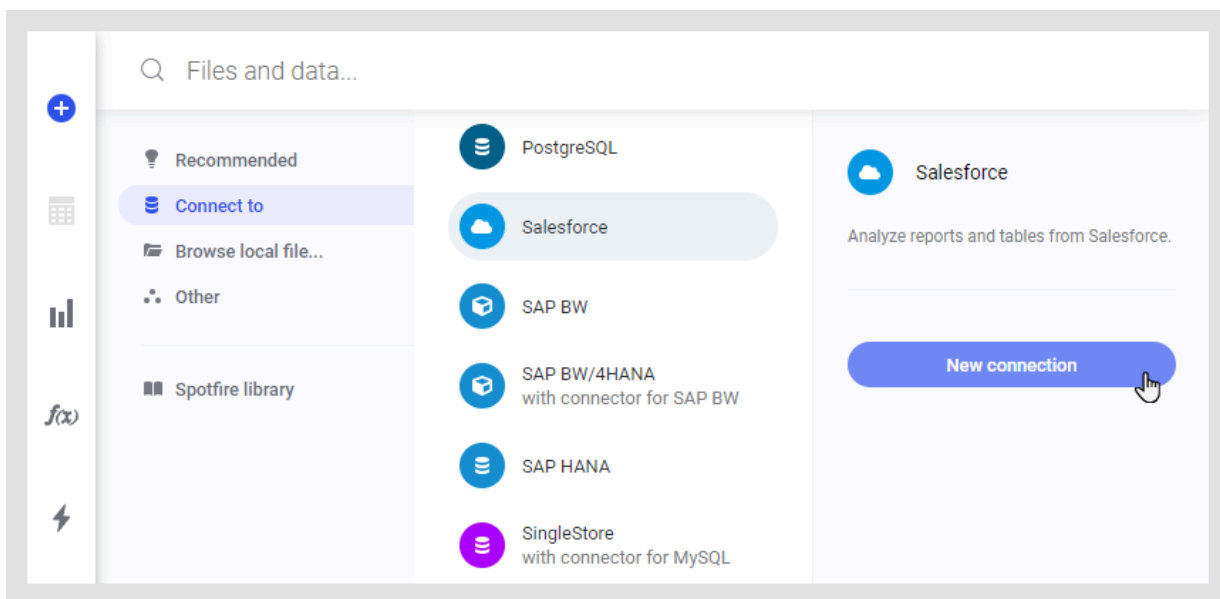
When you add data from an information link, you can only import the data, for in-memory analysis.

When it is possible, it is recommended to use data connections rather than information links to access data.

## Supported data sources

The following are the external data sources that this version of Spotfire has native support for. In order to connect to a data source, you use a connector to create a data connection. Some connectors have support for more than one data source, which is why there are more supported data sources than connectors.

All these data sources are available in the **Connect to** list in the **Files and data**  flyout. From there you can easily create a new connection or find shared connections in your library.



## Data sources and connectors

For help and documentation about connecting to and accessing data from a specific data source, follow the link for the corresponding connector in the list.

Data source	Connector name
Amazon Aurora MySQL	<a href="#">MySQL</a>
Amazon Aurora PostgreSQL	<a href="#">PostgreSQL</a>
Amazon EMR Hive	<a href="#">Hortonworks</a>
Amazon EMR Spark SQL	<a href="#">Apache Spark SQL</a>
Amazon RDS for MariaDB	<a href="#">MySQL</a>
Amazon RDS for MySQL	<a href="#">MySQL</a>
Amazon RDS for Oracle	<a href="#">Oracle</a>
Amazon RDS for PostgreSQL	<a href="#">PostgreSQL</a>
Amazon RDS for SQL Server	<a href="#">Microsoft SQL Server</a>
Amazon Redshift	<a href="#">Amazon Redshift</a>
Apache Drill	<a href="#">Apache Drill</a>
Apache HAWQ	<a href="#">Apache HAWQ</a>
Apache Hive	<a href="#">Hortonworks</a>
Apache Spark SQL	<a href="#">Apache Spark SQL</a>
Attivio	<a href="#">Attivio</a>

Data source	Connector name
Brytlyt	<a href="#">PostgreSQL</a>
Cloudera Hive	<a href="#">Cloudera Hive</a>
Cloudera Impala	<a href="#">Cloudera Impala</a>
Cloudera Spark SQL	<a href="#">Apache Spark SQL</a>
Databricks	<a href="#">Apache Spark SQL</a>
Dremio	<a href="#">Dremio</a>
Google Analytics	<a href="#">Google Analytics</a>
Google BigQuery	<a href="#">Google BigQuery</a>
Google Cloud SQL for MySQL	<a href="#">MySQL</a>
Google Cloud SQL for PostgreSQL	<a href="#">PostgreSQL</a>
Google Cloud SQL for SQL Server	<a href="#">Microsoft SQL Server</a>
Greenplum	<a href="#">Greenplum</a>
Hortonworks Hive	<a href="#">Hortonworks</a>
Hortonworks Spark SQL	<a href="#">Apache Spark SQL</a>
IBM BigInsights Hive	<a href="#">Hortonworks</a>
IBM BigInsights Spark SQL	<a href="#">Apache Spark SQL</a>
IBM Db2	<a href="#">IBM DB2</a>
IBM Db2 Big SQL	<a href="#">IBM DB2</a>
IBM Netezza	<a href="#">IBM Netezza</a>
MapR Hive	<a href="#">Hortonworks</a>
MapR Spark SQL	<a href="#">Apache Spark SQL</a>
MariaDB	<a href="#">MySQL</a>
Microsoft Azure Analysis Services	<a href="#">Microsoft SQL Server Analysis Services</a>
Microsoft Azure Database for MariaDB	<a href="#">MySQL</a>
Microsoft Azure Database for MySQL	<a href="#">MySQL</a>
Microsoft Azure Database for PostgreSQL	<a href="#">PostgreSQL</a>
Microsoft Azure HDInsight Hive	<a href="#">Hortonworks</a>
Microsoft Azure HDInsight Spark SQL	<a href="#">Apache Spark SQL</a>

Data source	Connector name
Microsoft Azure SQL Database	<a href="#">Microsoft SQL Server</a>
Microsoft Azure Synapse Analytics	<a href="#">Microsoft SQL Server</a>
Microsoft SQL Server	<a href="#">Microsoft SQL Server</a>
Microsoft SQL Server Analysis Services	<a href="#">Microsoft SQL Server Analysis Services</a>
Microsoft SharePoint Online	<a href="#">Microsoft SharePoint Online</a>
MongoDB Connector for BI	<a href="#">MySQL</a>
MySQL	<a href="#">MySQL</a>
OData	<a href="#">OData</a>
Oracle	<a href="#">Oracle</a>
Oracle Essbase	<a href="#">Oracle Essbase</a>
PostgreSQL	<a href="#">PostgreSQL</a>
SAP BW	<a href="#">SAP BW</a>
SAP BW/4HANA	<a href="#">SAP BW</a>
SAP HANA	<a href="#">SAP HANA</a>
Salesforce	<a href="#">Salesforce</a>
SingleStore	<a href="#">MySQL</a>
Snowflake	<a href="#">Snowflake</a>
TIBCO Cloud Data Streams	<a href="#">Spotfire Data Streams</a>
TIBCO Cloud Live Apps	<a href="#">TIBCO Cloud™ Live Apps</a>
TIBCO ComputeDB	<a href="#">Apache Spark SQL</a>
TIBCO Data Virtualization	<a href="#">TIBCO Data Virtualization</a>
TIBCO Spotfire Advanced Data Services	<a href="#">TIBCO Data Virtualization</a>
Spotfire Data Streams	<a href="#">Spotfire Data Streams</a>
Teradata	<a href="#">Teradata</a>
Teradata Aster	<a href="#">Teradata Aster</a>
Vertica	<a href="#">Vertica</a>

## Accessing data from Apache Spark SQL and Databricks


You can access data from Spark SQL and Databricks systems in Spotfire.

To connect to data in Spark SQL or Databricks systems, you use the connector for Apache Spark SQL. To learn about the functionality and features available when you work with data from these systems, see [Connector for Apache Spark SQL — Features and settings](#) on page 91.

### Prerequisites

- The Apache Spark SQL connector requires a driver on the computer running Spotfire. See [Drivers and data sources in Spotfire](#).
- To make sure that your database is supported, see the [system requirements](#) for the Apache Spark SQL connector.

### Procedure

1. Open the **Files and data**  flyout, and click **Connect to**.
2. In the list of data sources, select **Apache Spark SQL** or **Databricks**.
3. In the panel on the right, choose if you want to create a new connection or add data from a shared data connection:
  - [Opening a shared data connection from the library](#)
  - [Create new connection](#)

## Working with and troubleshooting Apache Spark SQL data connections

The following is information specifically about working with data from an Apache Spark SQL connection in Spotfire.

### Prerequisite: Spark Thrift Server

To access data in Apache Spark SQL with the Spotfire connector for Apache Spark SQL, the Spark Thrift Server must be installed on your cluster. Spark Thrift Server provides access to Spark SQL via ODBC, and it might not be included by default on some Hadoop distributions.

### Prerequisite: `spark.shuffle.service.enabled`

If you use the in-database load method when connecting to Apache Spark 2.1 or later, and you encounter errors in your analysis, the option `spark.shuffle.service.enabled` might have to be enabled on the Spark server.

### Connecting to Databricks SQL Analytics

You can also create an Apache Spark SQL connection for performing Databricks SQL Analytics queries. To be able to connect to Databricks, you must install the Databricks ODBC driver. Check the [system requirements](#) for the Apache Spark SQL connector, and see [Drivers and data sources in Spotfire](#) for finding the right driver.

### Databricks cluster that is not running

When connecting to a Databricks cluster that is not already running, the first connection attempt will trigger the cluster to start. This can take several minutes. The **Database** selection menu will be populated once Spotfire is connected successfully. You may have to click **Connect** again if the connection times out.

### Apache Spark SQL temporary views and tables in custom queries

If you are creating a custom query and you want to use data from an Apache Spark SQL temporary table or view, you must refer to those objects using their qualified names, specifying both the name and the location of the object. The qualified names required have the following format:

```
databaseName.tempViewName
```

By default, global temporary views are stored in the `global_temp` database. The database name can vary, and you can see it in the hierarchy of available database tables in Spotfire. To select all columns from a global temporary view named `myGlobalTempView`, that is stored in the `global_temp` database:

```
SELECT * FROM global_temp.myGlobalTempView
```

Temporary views/tables (listed in Spotfire under 'Temporary views' or 'Temporary tables') are always located in the `#temp` database. To select all columns in a temporary view named `myTempView`:

```
SELECT * FROM #temp.myTempView
```

### User agent tagging

If the ODBC driver that you use supports the `UserAgentEntry` option, Spotfire includes the following string as the `UserAgentEntry` in queries:

```
TIBCOSpotfire/<ProductVersion>
```

## Connector for Apache Spark SQL — Features and settings

You can connect to and access data from Spark SQL databases and Databricks with the data connector for Apache Spark SQL. On this page, you can find information about the capabilities, available settings, and things to keep in mind when you work with data connections to Apache Spark SQL.

### Page content

- [Connector features](#) on page 91
- [Data source properties](#) on page 92
- [Supported data types](#) on page 94
- [Supported functions](#) on page 94

### Connector features

The following functionality is available when you access data with the connector for Apache Spark SQL.

Feature	Supported?
Load methods	<ul style="list-style-type: none"> <li>• Import (in-memory)</li> <li>• External (in-database)</li> <li>• On-demand</li> </ul>
Custom queries	Yes
Stored procedures	Yes
Custom connection properties	Yes
Single sign-on with identity provider	Yes


Feature	Supported?
Authoring in web client	Yes
Supported on Linux Web Player	Yes

## Data source properties

The following are the supported data source properties that you can configure when you create a data connection with the connector for Apache Spark SQL.

Option	Description
Server	<p>The name of the server where your data is located.</p> <p>To include the port number that the Spark Thrift Server listens on, add it directly after the name preceded by colon.</p> <p>Example: MyDatabaseServer:10001</p> <p>Default port number: 10000</p>
Authentication method	<p>The authentication method to use when logging into the database. The following options are available:</p> <ul style="list-style-type: none"> <li>• No authentication</li> <li>• Kerberos</li> <li>• Username</li> <li>• Username and password</li> <li>• Microsoft Azure HDInsight Service</li> <li>• Identity provider (OAuth2)</li> </ul>
Host FQDN	<p>[Only available for Kerberos authentication.]</p> <p>The fully qualified domain name of the Spark Thrift Server host. For more information about the host FQDN, contact your Apache Spark SQL system administrator.</p>
Service name	<p>[Only available for Kerberos authentication.]</p> <p>The Kerberos service principal name of the Spark server. For example, "spark". For more information about the service name, contact your Apache Spark SQL system administrator.</p>
Realm	<p>[Only available for Kerberos authentication.]</p> <p>The realm of the Spark Thrift Server host. Leave blank if a default Kerberos realm has been configured for your Kerberos setup. For more information about the realm, contact your Apache Spark SQL system administrator.</p>
Identity provider	<p>[Only applicable for Identity provider (OAuth2) authentication.]</p> <p>Select the identity provider you want to use for logging in to the data source. The options available in the drop-down menu are the identity providers you have added to the OAuth2IdentityProviders preference.</p>



Option	Description
Scopes	<p>[Only applicable for Identity provider (OAuth2) authentication.]</p> <p>Scopes determine what permissions Spotfire requests on your behalf when you log in to the data source.</p> <p><b>Default</b></p> <p>Use the default scopes that you have specified for your identity provider in the <code>OAuth2IdentityProviders</code> preference.</p> <p><b>Custom</b></p> <p>Enter scopes manually in the text box. Separate values with a space.</p> <pre>Scope_1 Scope_2</pre>
Thrift transport mode	<p>Select the transport mode that should be used to send requests to the Spark Thrift Server. The following options are available:</p> <ul style="list-style-type: none"> <li>• Default (The Spark SQL ODBC driver will use either binary or SASL, depending on the Spark Server version you are connecting to.)</li> <li>• Binary</li> <li>• SASL</li> <li>• HTTP</li> </ul>
HTTP Path	<p>[Only available for <b>Thrift transport mode HTTP</b>.]</p> <p>Specify the partial URL that corresponds to the Spark server you are connecting to.</p> <div>  <p>The partial URL is appended to the host and port specified in the <b>Server</b> field.</p> <p>For example, to connect to the HTTP address <code>http://example.com:10002/gateway/default/spark</code>, you would enter the following:</p> <pre>Server: example.com:10002 HTTP Path: /gateway/default/spark</pre> </div>
Connection timeout (s)	The maximum time, in seconds, allowed for a connection to the database to be established. The default value is 120 seconds.
Command timeout (s)	The maximum time, in seconds, allowed for a command to be executed. The default value is 1800 seconds.

## Custom properties for Apache Spark SQL connection data sources

The following is the default list of driver settings that are allowed as custom properties in Apache Spark SQL connection data sources. To learn how to change the allowed custom properties, see [Controlling what properties are allowed](#).


### Default allowed custom properties

```
ADUserNameCase, AOSS_AuthMech, AOSS_CheckCertRevocation, AOSS_Min_TLS, AOSS_PWD,
AOSS_TrustedCerts,
AOSS_UID, AOSS_UseSystemTrustStore, AsyncExecPollInterval, AutoReconnect,
BinaryColumnLength,
Canonicalization, CheckCertRevocation, ClientCert, ClientPrivateKey,
ClientPrivateKeyPassword,
ClusterAutostartRetry, ClusterAutostartRetryTimeout, DecimalColumnScale,
DefaultStringLength,
DelegateKrbCreds, DelegationUID, DriverConfigTakePrecedence, EnableAsyncExec, EnablePKFK,
EnableQueryResultDownload, EnableStragglerDownloadMitigation,
EnableSynchronousDownloadFallback,
```

```
FastSQLPrepare, ForceSynchronousExec, HTTPAuthCookies, InvalidSessionAutoRecover,
LCaseSspKeyName,
MaximumStragglersPerQuery, Min_TLS, ProxyHost, ProxyPort, ProxyPWD, ProxyUID,
QueryTimeoutOverride,
RateLimitRetry, RateLimitRetryTimeout, RowsFetchedPerBlock, ServiceDiscoveryMode,
ShowSystemTable,
SocketTimeout, StragglerDownloadMultiplier, StragglerDownloadPadding,
StragglerDownloadQuantile,
ThrowOnUnsupportedPkFkRestriction, TrustedCerts, TwoWaySSL, UseNativeQuery, UseOnlySSPI,
UseProxy,
UseSystemTrustStore, UseUnicodeSqlCharacterTypes
```

## Supported data types

When you are setting up a connection to an external data source, Spotfire needs to map the data types in the data source to data types in Spotfire. The following are the data types that the Apache Spark SQL connector supports.

Database data type	Spotfire data type
BINARY	Binary
BOOLEAN	Boolean
TIMESTAMP	DateTime
TINYINT	Integer
SMALLINT	Integer
INT	Integer
BIGINT	LongInteger
DOUBLE	Real
FLOAT	SingleReal
STRING	String
DECIMAL (precision (p), scale (s))	When p = 0 and s = 0: Currency When p ≤ 9 and s = 0: Integer When p ≤ 18 and s = 0: LongInteger When p ≤ 15: Real Else: Currency  DECIMAL columns from temporary tables/views are always mapped to the Spotfire data type Currency, because their precision (p, s) is unlimited (0, 0).

## Supported functions

Supported functions are the functions that you can use when you work with in-database data tables, for example for calculated columns and custom expressions.



Some supported functions might not be possible to use with your database. This depends on what functions are available in the database, which often differs between database versions and types.

The following are the functions that the Apache Spark SQL connector supports.

Function type	Functions supported
Date and Time	DateDiff, Date_Add, Date_sub, Day, DayOfMonth, From_utc_timestamp, Hour, Minute, Month, Quarter, Second, To_date, To_utc_timestamp, Week, WeekOfYear, Year
Conversion	SN
Math	Abs, ACos, ASin, Atan, Bin, Ceil, Ceiling, Conv, Cos, Degrees, E, Exp, Floor, Hex, Ln, Log, Log2, Log10, Negative, Pi, Pmod, Positive, Pow, Power, Radians, Rand, Round, Sign, Sin, Sqrt, Tan
Operators	%, +, -, *, /
Statistical	Avg, Bit_And, Bit_Or, Bool_And, Bool_Or, Corr, Count, Covar_pop, Covar_samp, Max, Min, Percentile, StdDev_Pop, StdDev_Samp, Sum, UniqueCount, Variance, Var_Pop, Var_Samp
Text	ASCII, Concat, Concat_ws, Find_in_set, Get_json_object, Instr, Length, Locate, Lower, Lcase, LPad, LTrim, Parse_url, Regexp_extract, Regexp_replace, Repeat, Reverse, RPad, Rtrim, Space, Translate, Trim, Ucase, Upper

### Other supported functionality

- Temporary view / Temporary tables
- Global temporary views



Binning is not supported by this connector.

## Accessing data from Microsoft SQL Server

You can connect to and access data from Microsoft SQL Server systems with the data connector for Microsoft SQL Server. On this page, you can find information about the capabilities, available settings, and things to keep in mind when you work with data connections to Microsoft SQL Server.

### Page content

- [Connector features](#) on page 96
- [Data source properties](#) on page 96
- [Custom properties for Microsoft SQL Server connection data sources](#) on page 98
- [SQL Server data types](#) on page 99
- [Supported functions](#) on page 100
- [Working with Microsoft SQL Server data connections](#) on page 100

## Connector features

The following functionality is available when you access data with the connector for Microsoft SQL Server.

Feature	Supported?
Load methods	<ul style="list-style-type: none"> <li>• Import (in-memory)</li> <li>• External (in-database)</li> <li>• On-demand</li> </ul>
Custom queries	Yes
Stored procedures	Yes
Custom connection properties	Yes
Single sign-on with identity provider	Yes
Authoring in web client	Yes
Supported on Linux Web Player	Yes

## Data source properties

The following are the supported data source properties that you can configure when you create a data connection with the connector for Microsoft SQL Server.



For more information about the properties and the corresponding settings in the driver software, see the official documentation from [Microsoft](#) about the .NET data provider for SQL Server.

Option	Description
Server	<p>The name of the server where your data is located. To include a port number, add it directly after the name preceded by comma. To include an instance name, add it directly after the server name preceded by backslash.</p> <p>Example with port number: MyDatabaseServer,1234</p> <p>Example with instance name: MyDatabaseServer\InstanceName</p>
Database	<p>Determines how you select which database to access.</p> <p>Select from list</p> <p>Select from a list of all available databases in the Database drop-down menu. The Database drop-down menu is populated after you click Connect.</p> <p>Note: To use the Select from list option, you must have access to the default database in the data source. The default database is often called 'master'.</p> <p>Enter name</p> <p>Manually enter the name of the database that you want to access.</p>

Option	Description
Authentication method	<p>The authentication method to use when logging into the database.</p> <p><b>Windows authentication</b></p> <p>When using Windows authentication, e.g., Kerberos, the access token of the logged in user will be used. Users that have been given the appropriate access rights to SQL Server will be able to connect and read data.</p> <p>Domain credentials are not stored in the analysis file.</p> <p><b>SQL Server authentication</b></p> <p>With database authentication the authentication is done using a database user. Database credentials can be stored, unencrypted, as part of the analysis file, using a setting in the Data Source Settings dialog. If credentials are found in the analysis file they will be used to automatically authenticate against the database.</p> <p>If no credentials or credentials profiles are found in the analysis file all who open the file will be prompted for database credentials.</p> <p>Note that there will be no prompting for credentials if the credentials embedded in the analysis file fail.</p> <p><b>Active Directory - Integrated</b></p> <p>[This authentication method is not supported if you open the data connection in a Spotfire web client that is running on a Linux system.]</p> <p>If you are using Active Directory and your domain is federated with Azure Active Directory, you can connect to Microsoft Azure SQL Database and authenticate with your logged in Azure Active Directory credentials. Select the authentication method Active Directory - Integrated.</p> <p>With this authentication method, credentials cannot be saved in the analysis file.</p> <p><b>Active Directory - Password</b></p> <p>Enter your Azure Active Directory principal user name and password for authentication with Microsoft Azure SQL Database.</p> <p>Use Active Directory - Password authentication only to connect to Microsoft Azure SQL Database.</p> <p><b>Identity provider (OAuth2)</b></p> <p>Use an identity provider, such as Okta or Keycloak, to log in. When you open the data connection, a web browser window opens, and you log in with the identity provider's login procedure.</p>
Encrypt	<p>Select this check box to require that encryption is used when connecting to the SQL Server. Encrypt is selected by default.</p>
Trust server	<p>[Only applicable when Encrypt is selected.]</p> <p>If you want use encryption to connect to an SQL Server which does not have a verifiable server certificate (for example if it is using a self-signed certificate), you can select this check box to trust the server and connect without validation of the server certificate.</p>
Identity provider	<p>[Only applicable for Identity provider (OAuth2) authentication.]</p> <p>Select the identity provider you want to use for logging in to the data source. The options available in the drop-down menu are the identity providers you have added to the OAuth2IdentityProviders preference.</p>

Option	Description
Scopes	<p>[Only applicable for Identity provider (OAuth2) authentication.]</p> <p>Scopes determine what permissions Spotfire requests on your behalf when you log in to the data source.</p> <p><b>Default</b></p> <p>Use the default scopes that you have specified for your identity provider in the <code>OAuth2IdentityProviders</code> preference.</p> <p><b>Custom</b></p> <p>Enter scopes manually in the text box. Separate values with a space.</p> <pre>Scope_1 Scope_2</pre>
Connection timeout (s)	<p>The maximum time, in seconds, allowed for a connection to the database to be established.</p> <p>You can also set this timeout as a preference in the Administration Manager, which will be used if you don't set a timeout in the connection dialog.</p> <p>If you set a timeout in the connection dialog this value overrides the settings in the Administration Manager preference. If you specify a timeout neither in the connection dialog nor in the preference, the default value of 120 seconds will be used.</p> <p>Note: If you set the connection timeout to zero, it will be interpreted as no timeout. This means that there will be no upper limit for trying to execute the command. This is generally not recommended.</p>
Command timeout (s)	<p>The maximum time, in seconds, allowed for a command to be executed.</p> <p>You can also set this timeout as a preference in the Administration Manager, which will be used if you don't set a timeout in the connection dialog.</p> <p>If you set a timeout in the connection dialog this value overrides the settings in the Administration Manager preference. If you specify a timeout neither in the connection dialog nor in the preference, the default value of 1800 seconds will be used.</p> <p>Note: If you set the command timeout to zero, it will be interpreted as no timeout. This means that there will be no upper limit for trying to execute the command. This is generally not recommended.</p>

## Custom properties for Microsoft SQL Server connection data sources

The following is the default list of driver settings that are allowed as custom properties in Microsoft SQL Server connection data sources. To learn how to change the allowed custom properties, see [Controlling what properties are allowed](#).

### Default allowed custom properties

```
ApplicationIntent, Attestation Protocol, Column Encryption Setting, ConnectRetryCount,
ConnectRetryInterval, Enclave Attestation Url, Enlist, Failover Partner, FailoverPartnerSPN,
HostNameInCertificate, IPAddressPreference, Load Balance MultiSubnetFailoverTimeout, Max
Pool Size, Min Pool Size, MultipleActiveResultSets, , Packet Size, PoolBlockingPeriod,
Pooling, ServerCertificate, ServerSPN, Transaction Binding, Type System Version, User
Instance, Workstation ID
```

## SQL Server data types

When you are setting up a connection to an external data source, Spotfire needs to map the data types in the data source to data types in Spotfire. See below for a list of the different data types that Spotfire supports and the applied data type mappings when working with a SQL Server database.

SQL Server data type	Spotfire data type
BINARY	Binary
GEOGRAPHY	Binary
GEOMETRY	Binary
IMAGE	Binary
ROWVERSION	Binary
TIMESTAMP	Binary
VARBINARY	Binary
BIT	Boolean
DECIMAL	Currency
MONEY	Currency
NUMERIC	Currency
SMALLMONEY	Currency
DATE	Date
DATETIME	DateTime
DATETIME2	DateTime
SMALLDATETIME	DateTime
INT	Integer
SMALLINT	Integer
TINYINT	Integer
BIGINT	LongInteger
FLOAT	Real
REAL	SingleReal
CHAR	String
NCHAR	String
NTEXT	String

SQL Server data type	Spotfire data type
NVARCHAR	String
TEXT	String
UNIQUEIDENTIFIER	String
VARCHAR	String
TIME	Time

## Supported functions

Supported functions are the functions that you can use when you work with in-database data tables, for example for calculated columns and custom expressions.



Some supported functions might not be possible to use with your database. This depends on what functions are available in the database, which often differs between database versions and types.

The following are the functions that the Microsoft SQL Server connector supports.

Function type	Functions supported
Date and Time	Year, Quarter, Month, Week, Day, DayOfMonth, DayOfYear, DayOfWeek, Hour, Minute, Second, Millisecond
Conversion	SN
Math	Abs, ASin, ACos, ATan, Atn2, Ceiling, Cos, Sin, Tan, Cot, Exp, Floor, Log, Log10, Power, Sqrt, Square
Operators	+, -, *, /, %
Statistical	Sum, Avg, Max, Min, Count, UniqueCount, Count_Big, Stdev, StdevP, Var, VarP
Text	CharIndex, Len, Lower, Upper, LTrim, RTrim, Soundex, Reverse, Split

## Working with Microsoft SQL Server data connections

The following is information specifically about working with data from a Microsoft SQL Server connection in Spotfire.

### Telemetry - Query tagging

When you establish a connection with the connector for Microsoft SQL Server, the Spotfire application name and version number are added to the connection string as a telemetry tag.

## Accessing data from Google Analytics

If you have a Google Analytics account, you can access and analyze data from Google Analytics in Spotfire.



### Prerequisites


You must have access to a Google Analytics account.



The Google Analytics connector must be enabled for use in web clients. For instructions on how to do this, see the topic *Enabling Google Connectors in Spotfire Web Clients* in the [Spotfire® Analyst User Guide](#).


## Procedure

1. On the authoring bar, click **Files and data** .
  2. In the Files and data flyout, click **Connect to** and, depending on the project you want to access, select one of the following options:
    - To access a project with Google Analytics 4 data, select **Google Analytics**.
    - To access a project with Universal Analytics data, select **Google Analytics (Universal Analytics)**.
-  To learn more, see [Google Analytics 4 or Universal Analytics](#) below.
3. In the Google Analytics flyout, click **New connection**.
    - You might have access to shared connections to Google Analytics in the library, which you can open to access a prepared selection of data from the external data source. Shared connections are listed under **In library**. If you access data from a shared connection, you just log in and click OK.
  4. Follow the login instructions in the dialogs that are displayed. That is, choose or add an account, provide credentials when necessary, and allow access.
  5. In the next dialog that is displayed, select which Google Analytics view you want to load data from. The selected view will become a data table in Spotfire.
  6. Click **Continue**.
  7. In the left part of the dialog, all the metrics and dimensions are grouped together in categories. Click on the plus sign next to the category name to expand a category and see the metrics and dimensions.
  8. Select the metrics and dimensions of interest.
 

 When you access Universal Analytics data, you can select a maximum of 10 metrics and 7 dimensions. For Google Analytics 4 data, the maximum is 10 metrics and 9 dimensions. This is due to a limitation in Google Analytics. Note that some combinations of metrics and dimensions will not be valid. For the latest information about which combinations are valid, see the official documentation for Google Analytics.

The selected metrics and dimensions are added to the list to the right in the dialog.
  9. Optionally, you can change what **Segment** to load data for. By default, the segment **All Users** is selected.
 

**System segments** are segments that are available to all users of the same account. **Custom segments** are segments that have been defined by yourself or your company.

 It is not possible to select segments when you access data from a Google Analytics 4 project.
  10. Optionally, you can change what **Date range** to load data for.
 

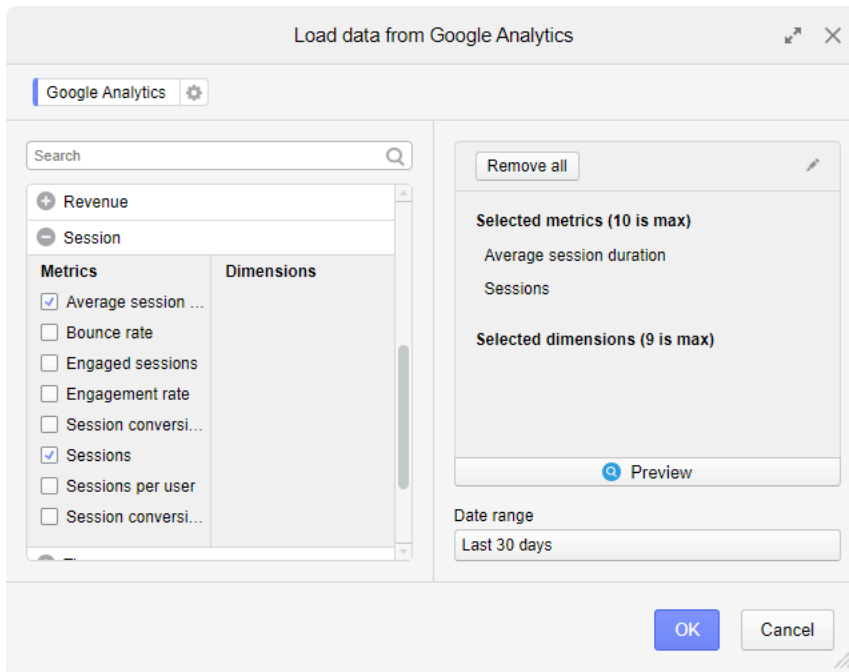
By default, the predefined range **Last 30 days** is selected, but you can change to another predefined range, specify your own range, or select to **Let the end user specify range**.
  11. Click **OK**.
 

The selected data is added to the summary view in the flyout.
  12. In the flyout, you can change the name of the new data table. Click **OK** when you are satisfied.
 

The selected metrics and dimensions are loaded into Spotfire.

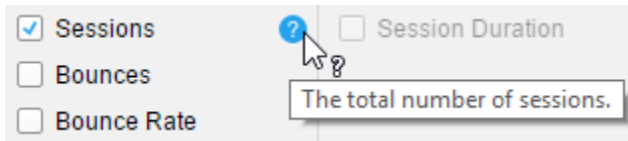
### The Load data from Google Analytics dialog

In this example, data will be loaded from My Account > My Property > My View 1. This data will become a data table when loaded into Spotfire. The left side of the dialog shows the available **Metrics** and **Dimensions** divided into categories. The category named **Session** has been expanded, and two metrics have been selected, **Sessions** and **Avg. Session Duration**. In the right part of the dialog, all the selected metrics and dimensions are listed; in this case four metrics and five dimensions. Each metric and dimension will become a column in the new data table. To rename columns, click on the pencil icon. In the lower right part of the dialog, **All Users** has been selected as the **Segment** and **Last 30 days** as the **Date range**.

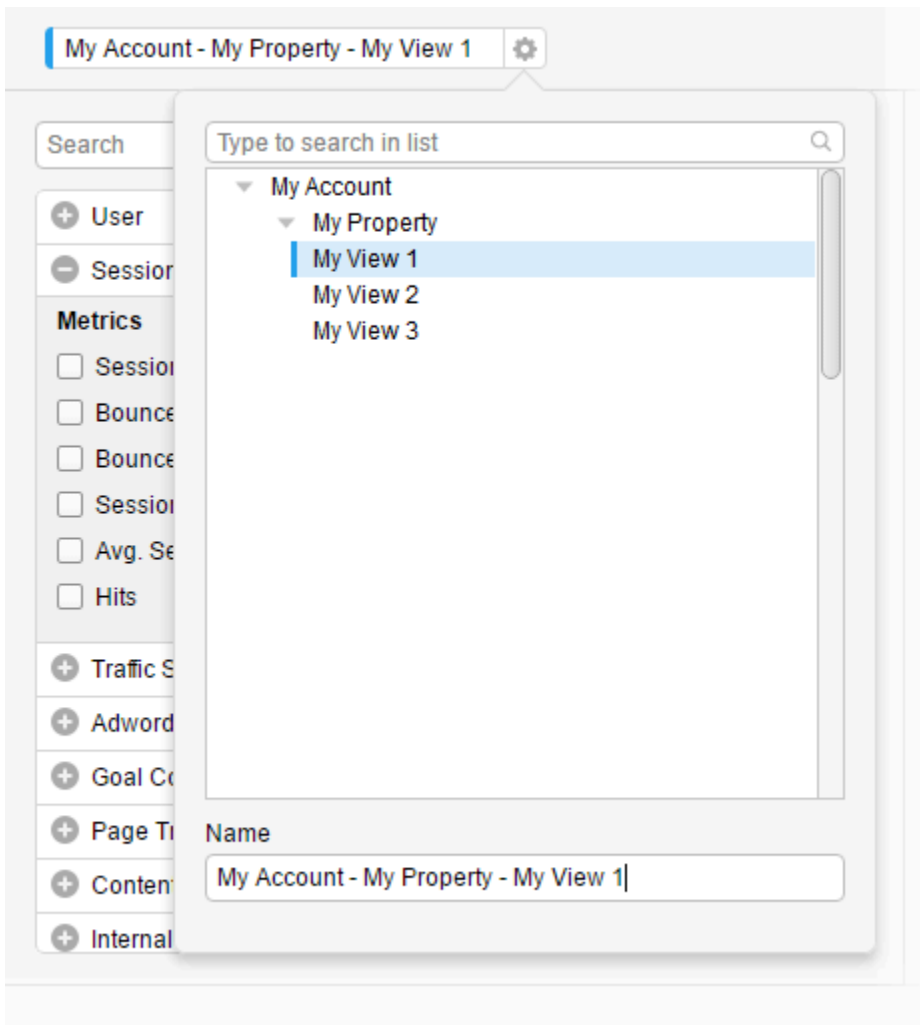


## Tips

To read a description, point to the question mark next to a metric or dimension name.



By default, the name of the data table is made up of the names of the selected account, property, and view. In this example, the name of the data table is My Account - My Property - My View 1. To change the name of the data table, click the settings icon to the right of the name. In the popover, you can change which view to load data from and you can rename the data table.



You can go back and edit the settings in this dialog from the Data canvas. See [Editing a data connection](#) on page 197 for more information.

## Google Analytics 4 or Universal Analytics

Spotfire supports accessing data from the current generation of Google Analytics, Google Analytics 4 (GA4), and from the legacy generation, Universal Analytics (UA). When you create a new data connection, in the **Connect to** list, select the data source that corresponds to your Google Analytics project:

- To access a project with GA4 data, select **Google Analytics**.



When you access GA4 data, Spotfire uses the libraries `Google.Analytics.Admin.V1Beta` and `Google.Analytics.Data.V1Beta` to call the Google Analytics APIs.

- To access a project with UA data, select **Google Analytics (Universal Analytics)**.

As an administrator, be aware that, technically, there are separate data connectors for GA4 and UA. This means that there is a license feature for each connector, which enables you to control users' access to the connectors. The connectors also have their technical own names in configuration files, used for enabling and configuring the connectors in web clients; `GA4Adapter` (for GA4) and `GoogleAnalyticsAdapter` (for UA).



If your project looks empty when you try to open it in Spotfire, the data in the project might be of the other type (UA or GA4). Try selecting the other Google Analytics data source in the **Connect to** list.


## Accessing data from Salesforce

If you have Salesforce account, you can access and analyze data from Salesforce in Spotfire.

### Prerequisites

You must have access to a Salesforce account.

### Procedure

1. On the authoring bar, click **Files and data** .
2. In the Files and data flyout, click **Connect to** and select **Salesforce**.
3. Click **New connection**.
  - You might have access to shared connections to Salesforce in the library, which you can open to access a prepared selection of data from the external data source. Shared connections are listed under **In library**. If you access data from a shared connection, you just log in and click OK.

4. In the Load data from Salesforce dialog, choose your preferred way to log in:

- **Log in with Salesforce**

This is typically the most convenient way to log in.

Click to launch the Salesforce login procedure in a separate web browser window. Follow the instructions for logging in, and then return to Spotfire.

- **Username and password (security token)**

If you want to log in directly in Spotfire, click to expand and display input fields for entering your username, password, and security token.

Enter your credentials and click **Log in**.



If you have a Salesforce sandbox that you want to connect to, select the **Sandbox** check box.

5. In the **Reports** and **Tables** lists, click to select which views to load.

Selected views are added to the middle of the dialog. Each selected view will be loaded into Spotfire as a new data table.

6. Click on a view in the middle of the dialog to see details for that specific view.



When you click on a view in the middle of the dialog, you will see all the columns that are included in that view. You can then choose which ones to include when the data is loaded. By default, all columns in a view are included.

7. Click **OK** when you have added the views of interest, and selected which columns to include in each view.

The selected data is added to the summary view in the flyout.

8. In the flyout, you can change the name of the new data table. Click **OK** when you are satisfied.

## Result

The selected data from Salesforce is loaded into Spotfire. You can configure your analysis so that the data is saved within the analysis instead of being reloaded each time the analysis is opened. See [Storing data within the analysis](#) to learn more about this.

## Prerequisites for using the 'Log in with Salesforce' authentication in your Spotfire web client

Using **Log in with Salesforce** is the most convenient way to connect to Salesforce in Spotfire, and it also enables you to use federated authentication with your own custom domain. If you want to use the Salesforce login procedure in your Spotfire web client, your Spotfire administrator must have made certain configurations:

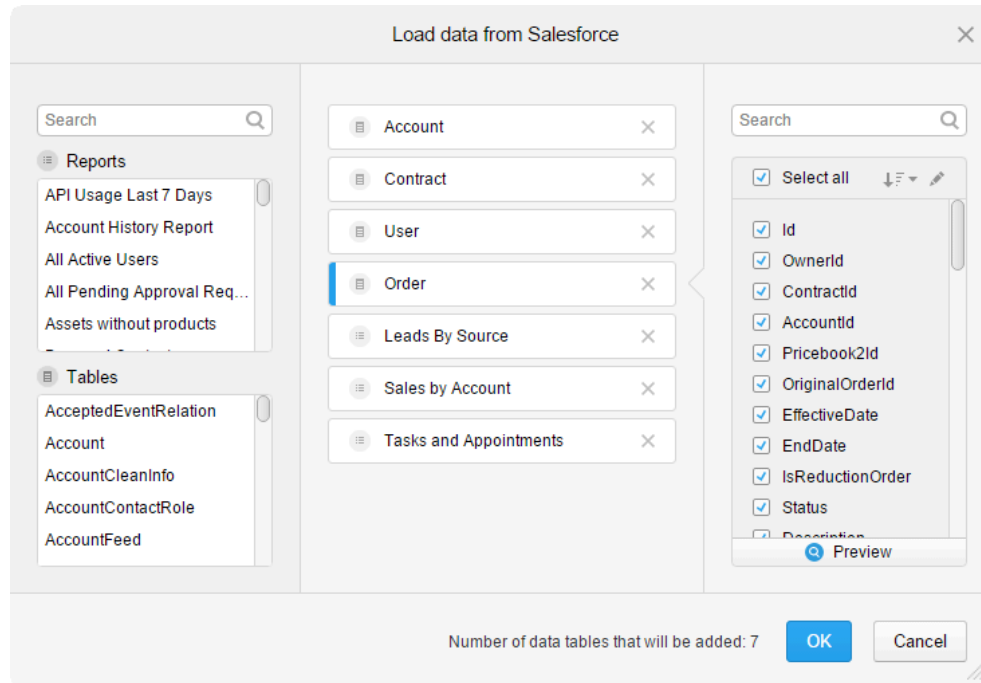
- Your Spotfire Server must be configured to use the HTTPS protocol.
- A Salesforce app must be created and configured to allow traffic from your Spotfire Server URL.



For more information, see the topic *Configuring the Salesforce connector for use in web clients* in the [Spotfire® Server Environment Installation and Administration Guide](#).

## The Load data from Salesforce dialog

In this example, seven views have been selected from the **Reports** and **Tables** lists. Three of them are reports and four of them are tables. The middle part of the dialog shows the added views in a list. The view named 'Order' has been selected in the middle part of the dialog, and to the right, all the columns included in the 'Order' view are shown. Exclude columns that you do not want to load by clearing their corresponding check boxes. To rename columns, click on the pencil icon.



You can go back and edit the settings in this dialog from the Data canvas. See [Editing a data connection](#) on page 197 for more information.

## Limitations when loading Salesforce data in Spotfire web clients

If the Salesforce instance you want to load data from contains very large amounts of data, keep the following limitations in mind when you create or edit a Salesforce connection in a Spotfire web client:

- If the Salesforce instance contains more than 1000 reports and tables, the search function in the Load data from Salesforce dialog is disabled.
- You might encounter issues when you select and add new data from Salesforce if the total number of columns in all selected reports or tables exceeds 2000. You can, however, create an analysis with such a connection in a Spotfire installed client and open it in the web client.

## Adding prompts to a Salesforce connection

When you create an analysis from Salesforce data, you can configure the analysis to let users select what data to load when opening the analysis.

Adding prompts to the analysis can be very useful as a way to narrow the amount of data that is loaded into Spotfire when the analysis is opened. By adding prompts to certain columns in the analysis, you

will let people opening the analysis select which values are of interest to them. For example, some people might find data from a specific geographical region interesting, while others do not. Instead of having to create many analyses with data specific to different regions, one single analysis can be created. Each person will select only the region that is relevant to them.




Prompts cannot be added to Salesforce reports, only to tables.

### Prerequisites

You must have access to a Salesforce account.

### Procedure

1. Follow the first steps in the help topic *Accessing data from Salesforce* or edit a Salesforce connection from the data canvas.
2. In the rightmost part of the Load data from Salesforce dialog, locate and point the mouse at the column of interest.  
This is the column from which you want end users to select values when they open the analysis.
3. Click on the icon next to the column name, .
4. Select **Let end user select values** from the pop-over menu.  
The pop-over is expanded with more settings.
5. Select which **Input type** to use. Read more in the separate section about Input types below.
6. Optionally, add a description to include in the prompt that will be shown when end users open the analysis.
7. Repeat steps 2 to 6 for each column that you want others to be able to select values from.
8. If you have added prompts to more than one column in the same view, you must consider the order in which the prompts will be displayed. Click on the **Input order** button.  
In the pop-over menu that is opened, all the columns with prompts are listed.
9. Click and drag the columns in the list to the order you want them to be displayed.
10. Click **OK**.  
The prompts you have added will be displayed in the order you specified in step 8.
11. For each prompt that is displayed, specify the values you want to load into Spotfire for further analysis.  
The selected data from Salesforce is loaded into Spotfire.

## Input types

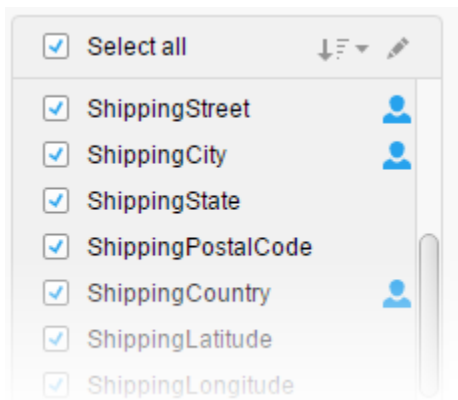
When you configure prompts for a column, you need to decide how to select values from the column.

- Use **Single selection** to only allow selection of one value from the column.
- Use **Multiple selection** to allow selection of more than one value from the column. The person using the connection will be presented with a list showing all the available values in the column. You can also specify how many values should be selected, using the **Min** and **Max** fields.
- Use **Range** to allow selection of multiple values between two values. The range option is most suitable for dates and columns containing numerical values. Note that if you use range as input type for categorical values, you must make sure you write good instructions in the **Description** field.
- Use **Manual input** to let the person using the connection enter any value to include (for string columns or numeric columns). This input type should only be used if all end users know which values are valid for the column, or if applicable values are shown in the description. For boolean values you get to pick a value using radio buttons (True/False) and for date columns you can choose a date from a calendar.

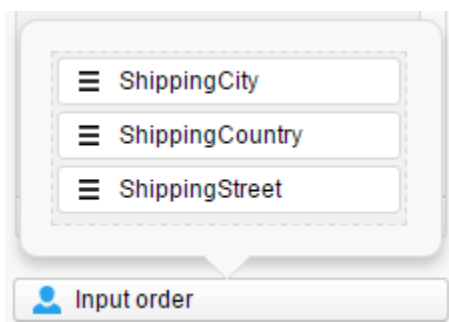


## Input order

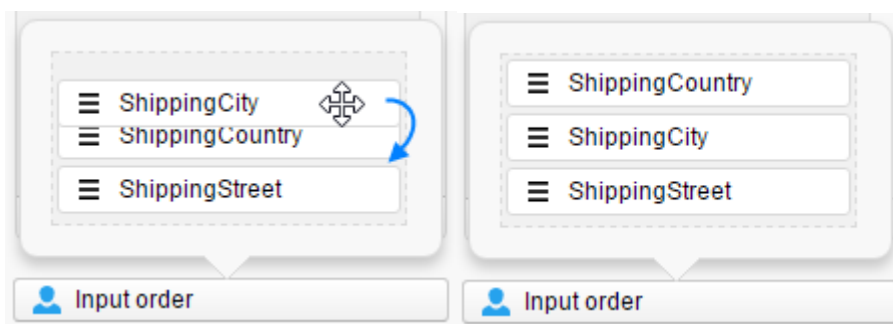
In this example, three columns have been configured with prompts: ShippingStreet, ShippingCity, and ShippingCountry.



The **Input order** pop-over shows the following order: ShippingCity, ShippingCountry, ShippingStreet.



A better order would probably be ShippingCountry, ShippingCity, ShippingStreet. That way the user would first specify which shipping countries to load data for. The next prompt would show only cities located in the selected countries. And the last prompt would show streets located in the selected cities. To change the order, click on a column and drag it to the position you prefer.





## Data connections and connection data sources

With a data connection, you can analyze data from an external system, such as a database, in Spotfire. To create and open data connections, you use the connector corresponding to the system you want to connect to. When you access data with data connections, you can either analyze data as in-database data tables, or import the data tables into the Spotfire data engine.

## What are data connections and connection data sources?

There are two fundamental components to data connections; the data connection itself, and the connection data source.

A **data connection**  stores information about what data in a database or cube that should be available as data tables in Spotfire. When you create a data connection, you define what data to include, and how that data should be shaped into one or more data tables.

A **connection data source**  stores information about how a data connection should access an external system. The information required to set up a connection data source varies between different connectors, but it typically includes a server name, port number, database name and credentials information.

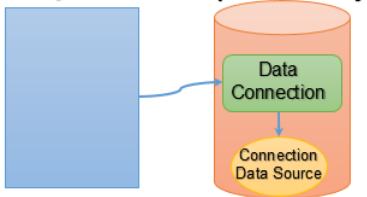
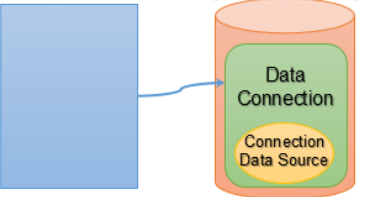
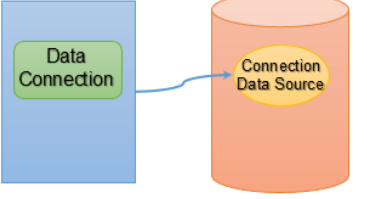
All data connections require a connection data source. The connection data source is often embedded in the data connection, and you can configure both the connection data source and the data connection when you create a new data connection.

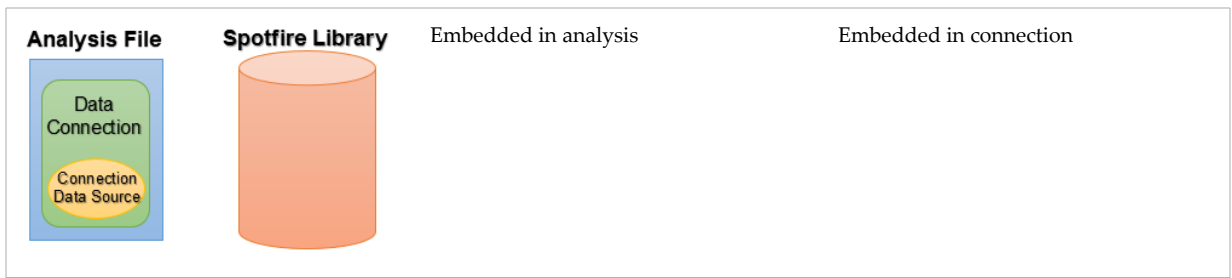
Both the connection data source and the data connection itself can be shared in the library, if desired. When you save a connection data source to the library, you can reuse that same connection data source to create new data connections.



You cannot use a data source specified in Information Designer as a connection data source.

The table below describes different configuration scenarios.

Spotfire analysis		Data connection	Connection data source
<b>Analysis File</b> 		Shared in library	Shared in library
<b>Analysis File</b> 		Shared in library	Embedded in connection
<b>Analysis File</b> 		Embedded in analysis	Shared in library



When your analyses use shared data connections you can easily update the data in a large number of analyses simultaneously by updating the data connection.

The data connections may either contain an embedded connection data source or a shared connection data source. When it comes to editing, a shared data source may temporarily be embedded in the connection, if necessary.

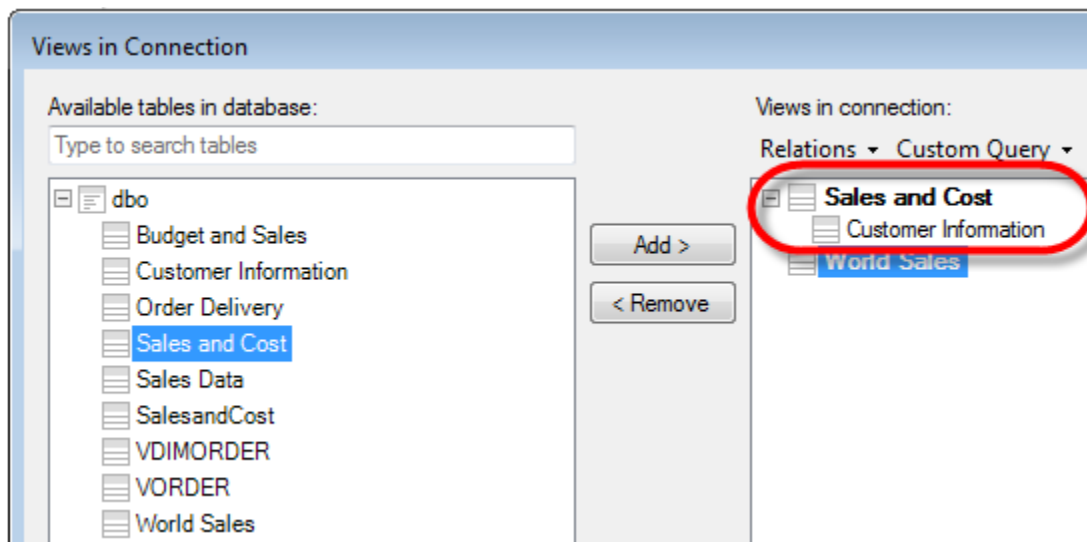
If your analysis contains a shared data connection you may still be able to use a cached version of the data connection if you are offline. However, this may result in inconsistencies with the expected data from the data connection on the library if the cached data is out of date.

### Views in the Connection

Data connections based on relational databases may contain one or more views, which may be used as data tables in the analysis. Each of the views may also be built by more than one of the source tables from the database, joined into a single view using structural relations. These relations may have been set up by the database administrator in the database but they may also be manually created in the Views in Connection dialog on the installed client.



A data connection from a cube data source always consists of a single view or data table, which combines all of the information from the selected cube, or part of the cube into one data table.



The selection of which data tables should be available in the analysis is also done in the Views in Connection dialog. In the image above, the data connection will create two views and one of those is a view combined by two related source tables. Data connections shared in the library contain a specified set of views, but data connections that are embedded in the current analysis can be edited within the context of the analysis and the currently used views in the connection can be updated. You can always embed a data connection in the Data Connection Settings dialog if you need to make any changes to the data tables in this particular analysis.

If you want to add data tables based on a view that is included in a connection already added to an analysis, you should add the data table from **Files and data > Recommended** option and open the data

connection listed under **Data connections in current analysis**, rather than adding a second connection based on the same source. Because the addition of a connection to an analysis adds all views available in that connection, you already have access to all of those views in the analysis and you only need to create a new data table from it. Adding a second connection based on the same data would create another set of views and, hence, increase the impact of the analysis on the external system.

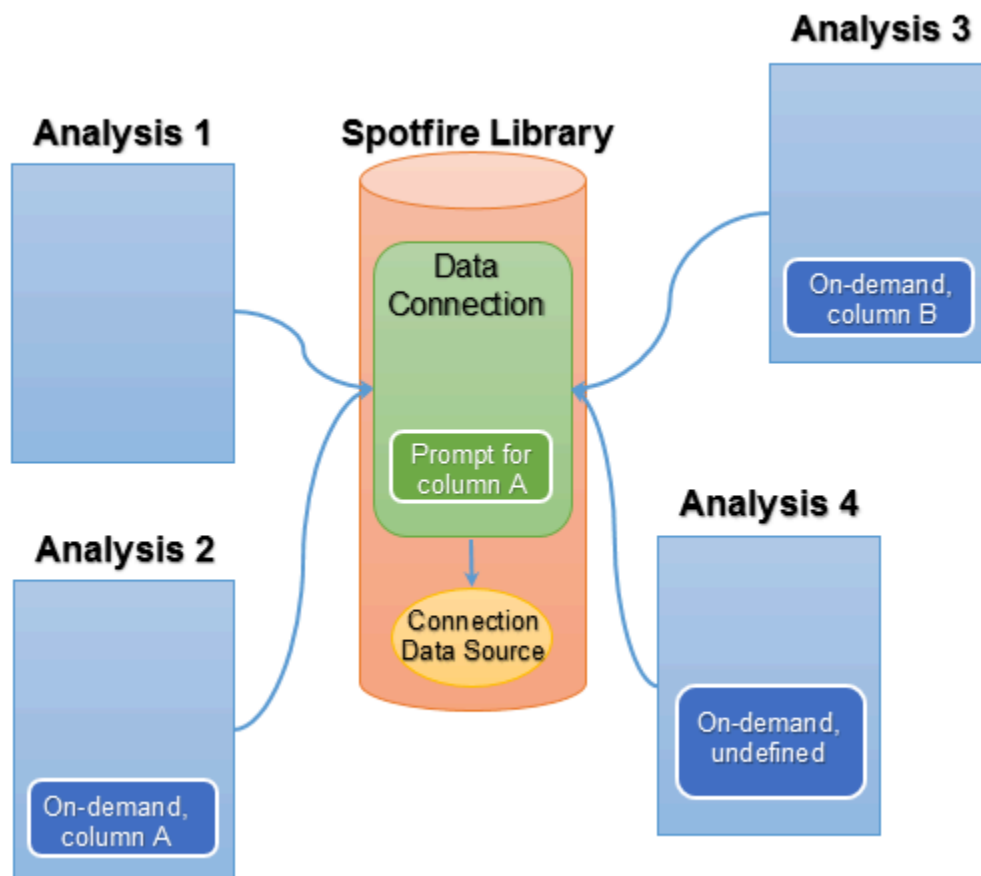
### Prompting and On-demand

Prompting and on-demand loading are two ways to give the end user of an analysis the chance to select which data from a data source to analyze. See [Loading Data Overview](#) for more information. When you define prompting for a data connection you always save the prompt configuration within the data connection itself. This means that all analyses that use a prompted data connection will request the end users for input when the analysis is opened.

However, on-demand loading is defined in the analysis. If you want to configure on-demand loading for an analysis based on a prompted data connection, then you can override the prompting by defining on-demand loading for the parameter which was configured with a prompt in the data connection.



Prompting and on-demand loading are available on the installed client.



In the image above, Analysis 1 prompts for input when the analysis is opened. It has no on-demand data tables at all, so the prompting is the only limitation done.

Analysis 2 has been configured with on-demand loading of the data from the data connection, and the same parameter that was configured with a prompt has been used to define on-demand loading (for example, selection is done by marking values in another data table). This means that the prompt is overridden and the input for parameter A will come from the defined on-demand input rather than from prompting.

Analysis 3 is configured to use on-demand loading, but the on-demand loading is based on input values for column B. In this case, the analysis will prompt for input to column A, and the limitations for the on-demand data table are solely based on selection of values for column B.

Analysis 4 is configured to use on-demand loading, but no on-demand parameters have been defined. In this case, only the prompting will take effect when data is loaded.

## Data connections in the Spotfire web client

When you use a Spotfire web client, you can access and analyze data from virtually any external data source that you can connect to in the installed client. Depending on the data source and method for connecting to it, you have access to different amounts of functionality in the web client.

For some data connectors, you have full authoring and access capabilities in the web client. This means you can create and edit connections to the corresponding data sources directly in the web client, to add data to your analysis.

For some other connectors, you only have the capability to access your data in the Spotfire web client. This means that connections to the corresponding data sources must be authored in an installed client. You can then continue to work on analyses that contain connections to these data sources in the web client. You can also open data connections that have been saved to the library, to add data to your analysis.

### Limitations when you author data connections in the web client

In the web client, you can open and use any data connection. However, to use some functionality, you must create the connection in the installed client.

The following data connection authoring capabilities are not available when you create or edit data connections in the web client:

- Writing custom queries
- Adding and editing prompting
- Selecting data from stored procedures
- Saving a data connection or connection data source to the library
- Manually creating structural relations

### Connectors with web client authoring support

Connector with web client authoring	Learn more
Apache Spark SQL	<a href="#">Connector for Apache Spark SQL — Features and settings</a> on page 91
Google Analytics	<a href="#">Accessing data from Google Analytics</a> on page 100
Microsoft SQL Server	<a href="#">Accessing data from Microsoft SQL Server</a> on page 95
Salesforce	<a href="#">Accessing data from Salesforce</a> on page 104

### Prerequisites for using connectors in the Spotfire web client

In the on-premises version of the Spotfire web client, most connectors that are available in the installed client can be used.

The connectors you want to use must be switched on in the Web Player service configuration, and the drivers must be installed on the computer running the Web Player service. Contact your Spotfire administrator if you need access to more connectors in the web client. For more information about deploying connectors, see [Setting up connectors](#) in the Spotfire Server and Environment Installation and Administration help.

## Information links

In a Spotfire installed client, you can use Information Designer to create information links that enable you to access data from JDBC-compliant data sources. When you use a Spotfire web client, you can open information links that you have saved to the library, and use analysis files that contain information links.

## Logging in to a data connection

When you open an analysis that contains data from a database, you typically have to log in to the external database. After logging in, you have access to up-to-date data directly from the external database in your analysis.

### Logging in

When you open an analysis with a data connection that requires that you log in, or when you add new data from a data connection to your analysis, a login dialog opens.

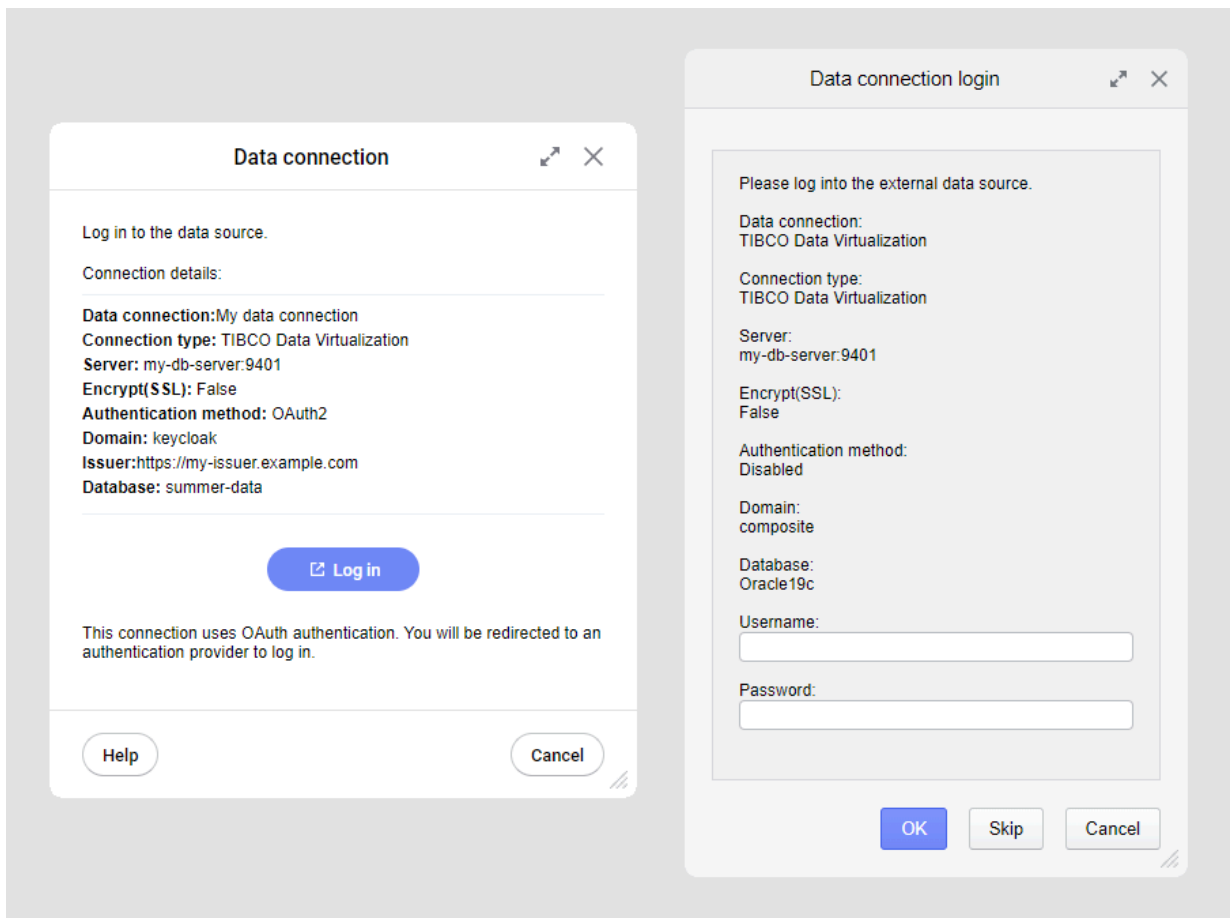
1. In the login dialog, view the details about the data connection and the external database. If the analysis contains multiple connections that you have to log in to, this information can be useful.
2. Depending on the type of connection and authentication method, enter your database credentials or click the login button to open the external login procedure.



If you cannot log in to a data connection when you open an analysis, for example if you do not have the credentials, you can click **Skip** to continue to open the analysis file, without loading data from the connection. Any data tables with data from the connection will be empty.

### Authentication methods and other settings

In Spotfire, data connections to databases can be configured in many ways that determine how you log in to the external database. When you open an analysis, the login dialog for the connection might look different for various reasons, for example if you are connecting to Google Analytics or Microsoft SQL Server.



Different authentication methods also affect what information you have to enter, and how. In some cases, you might not have to take any active steps at all to log in, if the connection is configured with an identity provider for authentication, or if the credentials are stored in a credentials profile.

## Properties in connection data sources

When you create or edit a data connection, you can set and change properties in the connection data source. Data source properties are settings that provide information about the external data source and decide how Spotfire should communicate with it.

Every data connector in Spotfire has a built-in set of data source properties that you can control. Some basic properties are available for most connectors, such as properties to specify the server address to connect to, the authentication method to use for login, and the username and password. Other properties are connector-specific, and some are applicable only for certain types of external data sources.

Usually, there is a match between a data source property in Spotfire and a setting or option in the driver software or API that Spotfire uses for communication with the external system. For example, if you create a connection with the Amazon Redshift connector, the data source property `SSL Mode` controls the option `SSLMode` in the Amazon Redshift ODBC driver. Therefore, to learn more about various data source properties, refer to the official documentation from the vendor of the database or driver.

## Configuring other driver settings as custom properties

With some connectors, you can add custom properties to your connection data source. With custom properties, you can add and change driver settings that do not have built-in data source properties. Custom properties can be useful for tweaking the connection performance, changing the behavior of a connection, or accessing certain data.

The following connectors support custom data source properties:

- [Apache Spark SQL](#)
- [Microsoft SQL Server](#)

## Adding a custom property to a data connection

To use a custom property in a data connection, you must add the property and set a value for it.



**Important** Before you add a setting as a custom property, refer to the official documentation from the database or driver vendor. You must know what the setting does, and what type of input it accepts. It is not certain that any particular driver setting is compatible with Spotfire as a custom property.



### Procedure

1. [Create a new data connection](#), or edit an existing one.
2. When you edit the connection data source in the **Connect to [data source]** form, click **More properties**.
3. In the expanded **More properties** form, click **Add property**.
4. In the popover menu, select the setting you want to add as a custom property.
5. In the new input field that was added, set a value for your custom property.

### Enabling custom properties for a connector

As a Spotfire administrator, for each data connector that supports custom properties, you can allow or turn off the use of custom properties. By default, custom properties are turned on.

#### Procedure

- To enable the use of custom properties for a connector, in the **Administration Manager**, set the preference `Connectors > [Connector Name] > AllowCustomProperties` to `True`.

### Controlling what custom properties are allowed

As a Spotfire administrator, you can enter a list of driver settings or options that are available as custom properties for each connector.



By default, and if the `CustomProperties` preference is empty, a list of default settings are allowed as custom properties for each connector. See the help topic for each connector for the full list.

#### Procedure

- To decide what custom properties are available for a connector, in the **Administration Manager**, edit the preference `Connectors > [Connector Name] > CustomProperties` and add a list of driver settings or options to allow.

## Single sign-on with an identity provider (OAuth2) for connectors

This article explains how to configure your Spotfire environment so that you can log in with an identity provider when you access data with connectors.

Some data connectors support using your own identity provider, such as Okta, Keycloak, or Google, for authentication. With such connectors you can provide a convenient log-in experience for users when they use a data connection or external library, without the hassle of having to remember separate database credentials. If you use the same identity provider for authentication in your Spotfire environment, you can even enable a full single sign-on experience.

### Prerequisites

- The following connectors support single sign-on with an identity provider:
  - Apache Spark SQL
  - Microsoft SQL Server
  - TIBCO® Data Virtualization
- Your identity provider must use one of the following protocols:
  - OAuth2
  - OpenID Connect
- You have configured the external system that you want to access data from to use your identity provider for authentication.
- You have registered client applications for Spotfire with your identity provider, one confidential and one public. The public client application is required for the Spotfire installed client, and the confidential client is required for Spotfire Server and Spotfire web clients.

### Is my identity provider supported?

Spotfire's support for authentication with an identity provider is built to be generic, and it is not tailored to any specific identity provider solution. An important prerequisite is that the identity provider is supported by the external data source.

The following are details about the Spotfire implementation, to help you understand the requirements and limitations on your identity provider:

- The identity provider must expose an OAuth2 or OpenID Connect authorization server, and Spotfire uses the Authorization Code Grant Flow to get an access token from the authorization server.
- Spotfire does not support adding custom headers in the authorization request.
- Spotfire does not support adding custom query parameters in the authorization request.

## Configuring Spotfire to use an identity provider for connectors

To be able to use an identity provider to log in to an external system from Spotfire, you must add the details about your identity provider.

There are 2 places in the Spotfire configuration where you must add your identity provider information; in the OpenID Connect settings on the Spotfire Server, where you use your confidential client application details, and in the preference `OAuth2IdentityProviders` in the Administration Manager, where you use your public client application details. For use with data connectors, it is important that you configure both, so that you can log in and access your data both in the Spotfire web client and in the Spotfire installed client.

In broad strokes, you must perform the following steps to complete the configuration:

1. Collect information about your identity provider, and both your client applications for Spotfire.
2. Add your identity provider on your Spotfire Server, with your confidential client application details.
3. Add your identity provider to the `OAuth2IdentityProviders` preference, with the public client application details.
4. Use your identity provider for authentication in data connections or external libraries.

### Collecting information about your identity provider

Before you start, collect the following information about your identity provider:

- What protocol does it use; `OpenID Connect` or `OAuth2`
- The `issuer identifier URL`
- Details about the client applications you have registered for Spotfire with your identity provider; `client ID` and `client secret` (if applicable)
- The scopes for the permissions required for accessing data in the external system

Depending on your identity provider, additional details might be required. For information about all the available settings, see the reference documentation about the [OAuth2IdentityProviders preference in the Administration Manager User's Guide](#).

### Adding your identity provider on your Spotfire Server

To be able to use single sign-on, and also to be able to use your data connections in Spotfire web clients, you must add your identity provider and the confidential client application information to the Spotfire Server. Depending on your use case, you have two options:

- To use the identity provider for authentication on the Spotfire Server and for data access with connectors, add the identity provider to the `OpenID Connect` settings on the Spotfire Server. See [Configuring OpenID Connect](#).
- To use the identity provider for authentication only for data access with connectors, add the identity provider on the Spotfire Server with the `config-oauth-client` command.

### Adding your identity provider to the OAuth2IdentityProviders preference

1. Start Spotfire Analyst, and log in as a user with administrator privileges.
2. On the menu bar, select **Tools > Administration manager...**
3. In the Administration Manager dialog, on the **Preferences** tab, click to select the user group you want to edit preferences for.
4. On the **Preferences** tab, click **Edit**.
5. In the Edit Preferences dialog, navigate to the preference **Application > OAuth2Preferences > OAuth2IdentityProviders**.
6. To edit the `OAuth2IdentityProviders` preference, select the preference and click the edit button [...].

7. In the String Collection Editor dialog, add your identity provider and the public client details as a JSON object.



**Important** Make sure to use the public client application from your identity provider, and not the confidential client application.

You can add your details to the sample below, which contains commonly used settings:

```
[
  {
    type: "[OAuth2 or OpenId]",
    displayName: "[My Identity Provider]",
    issuer: "[https://issuer1.example.com]",
    publicClient: {
      id: "[Client name or ID]",
      redirectUrl: "[redirect-port]",
      redirectPorts: "[port-number]"
    },
    defaultScope: "myScope"
  }
]
```

8. To save your changes, click **OK**.



For more information about all the possible settings, see the reference documentation about the [OAuth2IdentityProviders preference in the Administration Manager User's Guide](#).

## Using your identity provider in data connections and external libraries

### Creating a data connection with your identity provider for authentication

To use your identity provider for authentication in a data connection, create a new data connection or connection data source and select the authentication method **Identity provider (OAuth2)**. You can select your identity provider (listed with its display name from the `OAuth2IdentityProviders` preference) in the **Identity providers** drop-down menu.

### Using your identity provider for authentication with an external library

Some connectors support both authentication with an identity provider and configuring an external library. In such cases, you can use identity providers that you have added to Spotfire for logging in to the external library.

When you set up an external library in the External library configurations, add the following settings:

```
authenticationMethod = "OAuth2"
issuer = "[Issuer URL of your Identity Provider]"
```

For more information, see [Configuring the TIBCO Data Virtualization Integration](#).

## Connecting to Data from TIBCO Data Virtualization (TDV)

With the TIBCO Data Virtualization integration in Spotfire, you can search and view data from TDV directly in the Files and Data flyout.

[The information on this page is only applicable if you use an on-premises version of Spotfire®. If you are using Spotfire® on TIBCO Cloud™, you cannot access data from TIBCO Data Virtualization with the method described below. You must use the [Spotfire Connector for TDV](#) instead.]

After you have configured the integration, you have access to the data on the configured TDV server. To access the data from other TDV servers, you can use the [Spotfire Connector for TIBCO Data Virtualization](#).

See the official [TIBCO® Data Virtualization documentation](#) to learn how to work with TDV and how to tailor your data.

## Prerequisites

- You must [configure the TDV integration](#) by setting a preference in the Administration Manager preferences.
- You must use Kerberos to authenticate against the TDV server. Make sure you have [Kerberos installed and configured correctly](#).


## Adding data from TDV to a Spotfire analysis

In Spotfire, you can browse and select from all the databases, schemas, catalogs, and tables you have access to in TIBCO Data Virtualization. Your data is available directly in the **Files and data** flyout.



Should you have any issues or questions, take a look at the [Troubleshooting section](#) in the Spotfire Analyst User Guide.

### Procedure

1. Click  to open the **Files and data** flyout.
2. Select **Data catalog**.



Your system administrator can set a custom display name in the Administration Manager preferences.

3. Double-click a database from the list to open it. Similarly, navigate through the schemas and catalogs until you reach the table you want to add. Double-click the table to add it to the analysis.



There is no Views in Connection dialog to select rows or columns when using the TDV integration to add data; you always add the whole table to the analysis.

4. The final step is a summary view of the data you are about to add to your analysis. Here, you can choose how the data will be added. You can **Import** the data, keep it **External** (in-db analysis), or load it **On-demand**. To finish and add the data table, click **OK**.

## Data preparation in Microsoft Excel

Before loading a Microsoft Excel file into an analysis, it is important that the data spreadsheet is free from irrelevant information and has a good structure to prevent misinterpretation. Possible actions that can be done before loading data are removing contextual information and combining columns into one.

The tabular format of the data in an Excel spreadsheet will be represented as a data table in your analysis. The first row with data in the spreadsheet will be interpreted as names of the data columns in the table, and the following rows will be interpreted as data rows.

### Remove contextual information

The following illustration shows a spreadsheet containing some contextual information above the actual data table. This will cause misinterpretation of the data.

F12				fx	
	A	B	C	D	E
1	Report date:	04/05/2007		Manager name:	John Smith
2				Division:	Fruits
3					
4			Category	Date	Sales
5			Apples	01/05/2007	62
6			Pears	01/05/2007	42
7			Melons	01/05/2007	32
8			Apples	02/05/2007	45
9			Pears	02/05/2007	34
10			Melons	02/05/2007	20
11					

Remove any contextual information before loading the data, to get good results. In the sheet below, there is no contextual information before the actual data set, so it will therefore be interpreted correctly.

H7		fx				
	A	B	C	D	E	F
1						
2						
3						
4			Category	Date	Sales	
5			Apples	01/05/2007	62	
6			Pears	01/05/2007	42	
7			Melons	01/05/2007	32	
8			Apples	02/05/2007	45	
9			Pears	02/05/2007	34	
10			Melons	02/05/2007	20	
11						
12						

### Combine columns

The data can be organized in different ways, short/wide or tall/skinny, but still contain the same information. Often, it is easier to visualize data organized in a tall/skinny format. You can transform your data in Spotfire by [unpivoting columns](#), but if you are more comfortable with doing this work in Excel, it can also be done before uploading the data to Spotfire.

For example, the following Excel spreadsheet contains the numbers of sold entrance tickets for five different desks. The data is organized in a short/wide format, that is, the spreadsheet has many columns with similar data.

Category	Date	Desk1	Desk2	Desk3	Desk4	Desk5
Adult	5/1/2012	62	19	26	111	34
Child	5/1/2012	20	6	5	11	8
Senior	5/1/2012	102	47	42	6	49
Adult	5/2/2012	74	35	37	4	65
Child	5/2/2012	17	8	8	0	10
Senior	5/2/2012	122	39	47	3	50
Adult	5/3/2012	91	43	24	1	24
Child	5/3/2012	16	8	5	6	1
Senior	5/3/2012	99	35	34	40	55
Adult	5/4/2012	51	21	13	49	104
Child	5/4/2012	16	4	7	5	15
Senior	5/4/2012	64	29	29	51	80

In the spreadsheet below, the same data is organized in the tall/skinny format. The values from the desks have been combined in one column.

Category	Date	Desk	Tickets
Adult	5/1/2012	Desk5	34
Adult	5/1/2012	Desk4	111
Adult	5/1/2012	Desk3	26
Adult	5/1/2012	Desk2	19
Adult	5/1/2012	Desk1	62
Child	5/1/2012	Desk5	8
Child	5/1/2012	Desk4	11
Child	5/1/2012	Desk3	5
Child	5/1/2012	Desk2	6
Child	5/1/2012	Desk1	20
Senior	5/1/2012	Desk5	49
Senior	5/1/2012	Desk4	6
Senior	5/1/2012	Desk3	42
Senior	5/1/2012	Desk2	47
Senior	5/1/2012	Desk1	102
Adult	5/2/2012	Desk5	65
Adult	5/2/2012	Desk4	4
Adult	5/2/2012	Desk3	37
Adult	5/2/2012	Desk2	35
Adult	5/2/2012	Desk1	74

#### Put type row below name row

In spreadsheets, there is often a row that specifies the column data type. If your Excel file contains a row with data type information, make sure to place that row below the row with column names. This ensures that Spotfire can process the rows correctly.

ID	Item	Cost	Type
<i>Integer</i>	<i>String</i>	<i>Real</i>	<i>String</i>
1	Pineapple	95.3	Fruit
2	Pear	13.2	Fruit
3	Avocado	98	Vegetable
4	Apple	7.5	Fruit

## Searching the library

You can search for basic names of files and data in the **Files and data** flyout. However, you can also search for particular types of library items, find the files that were last updated, or find all files updated by a certain person. Other parts of the Spotfire environment provide even more opportunities to find specific items in the library.




This topic is not applicable for users running Spotfire® Analyst without a server. The concept of a library only applies to environments with a server.



Depending on where you are searching, you might get different search results, because only allowed types will be returned. For example, when searching in the Files and data flyout, only items that can be opened or pinned to a flyout, such as analyses, data connections, data functions, or mods are shown.





When browsing in the Files and data flyout, the content in the library can be sorted by **Name**, **Modified**, and **Type** by clicking  in the upper right corner.

Searching for a text string will by default look for matching text in the title and keywords of the items in the library. You can use wildcards and boolean operators to search for parts and combinations of words. For a listing of the basic search syntax, see [Searching in Spotfire clients](#) on page 956.

### Library-specific search

Keyword	Example	Function
title:<word in title>	title:sales	Locates library items with the specified word (or part of word) somewhere in the title.
created_by: <username>	created_by:admin createdby:admin created_by::admin	Locates library items created by a certain user.  In the first examples, all items modified by any users beginning with 'admin' will be found.  In the last example, only items modified by the user 'admin' will be found.
modified_by: <username>	modified_by:admin modifiedby:admin	Locates library items modified by a certain user.



Keyword	Example	Function
<code>item_type:&lt;type&gt;</code> or <code>type:&lt;type&gt;</code>	<code>item_type:datasource</code> <code>type:datafunction</code> <code>type:visualization</code> <code>type::visualizationmod</code> <code>type:action</code> <code>item_type::actionmod</code> <code>type:*mod</code>	<p>Locates items of a specific type.</p> <p> Some of the listed types are only available when searching in the Library Administration tool, or in the Spotfire Server Library page. If you use TIBCO Cloud™ Spotfire®, information model elements are not available.</p> <p>The available types in the files and data flyout are: <code>folder</code>, <code>dxp</code> (= Spotfire analysis file), <code>SBDF</code> (= Spotfire Binary Data Format file), <code>datafunction</code>, <code>dataconnection</code>, <code>actionmod</code> (= Spotfire action mod), and <code>visualizationmod</code> (= Spotfire visualization mod). You can also use the type <code>*mod</code> to find all kinds of mods in one search.</p> <p> In the Files and data flyout, you can search for <code>type:alldata</code> to limit the results to only include data connections, connection data sources, information links, data functions, and SBDF files.</p> <p>In other views you can also search for <code>connectiondatasource</code> (= a data source for a data connection), and <code>colorscheme</code>.</p> <p>If your company uses Information Services, you can also search for the following in certain views: <code>query</code> (=information link), <code>datasource</code> (= a data source in Information Services), and the Information Services elements <code>column</code>, <code>filter</code>, <code>join</code>, and <code>procedure</code>.</p>
<code>item_id::&lt;GUID&gt;</code> or <code>id::&lt;GUID&gt;</code>	<code>item_id::dac3cd8c-47ec-454a-a8f2-691c60ece052</code>	<p>Locates a specific library item based on its unique identifier.</p>
<code>description</code>	<code>description:sales</code> <code>type:dxp description:*sales*</code> <code>description::null</code>	<p>Locates all items containing the specified word in their description.</p> <p>Locates all analysis files containing the specified word in their description.</p> <p>Locates all items that are missing a description.</p> <p>Use wildcards (*) to locate items with the word anywhere in the description.</p>
<code>keywords</code>	<code>keywords:sales</code> <code>type:dxp keywords:*sales*</code>	<p>Locates all items containing the specified keyword.</p> <p>Locates all analysis files containing the specified keyword.</p> <p>Use wildcards (*) to locate items with the word anywhere in the keywords.</p>

Keyword	Example	Function
connector: <connector name> (this keyword is only available when searching in the <b>Data in analysis</b> flyout)	connector:sql  connector:"Microsoft SQL Server"	Locates all data connections or connection data sources from a certain connector.  Use the <a href="#">adapter</a> name or the full data connector name within quotes.
spotfire.connector: <connector name> (this can be used for all library searches)	spotfire.connector:google  spotfire.connector:Spotfire.GoogleAnalyticsAdapter	Locates all data connections or connection data sources from a certain connector, including those embedded in analyses.  Use the <a href="#">adapter</a> name or the beginning of the adapter name (the part after Spotfire.) to search.
depends_on (<expression> )	depends_on(item_id::538bcde4-7212-475f-a348-5bb41ba39c41)  depends_on(title:Sales)  depends_on(type::dataconnection)	Locates all items that depend on a specific element.  If the GUID in the example to the left belongs to a data connection, the search will find all analyses that use that data connection.
required_by (<expression> )	required_by(item_id::6f6dc7e0-57bd-11d7-5ac0-0010ac110132)  required_by(title:MyAnalysis)  required_by(type::dataconnection)	Locates all items that are required by another item.
modified	modified:"2 days ago"  modified:"a week ago"  modified:>"an hour ago"  modified:today  modified:<"this month"  modified::>created  modified:"2015-02-01T18:27:55CEST"	It is possible to search for items that have been modified during a specified time span, relative to today. There are two different ways of describing relative dates and times:  1) State the number of time parts ago in a string surrounded by quotes. The available time parts are seconds, minutes, hours, days, weeks, months and years. For example, search for modified:<"6 months ago". The given number of time units will be subtracted from the current time in the search.  2) State the time period to look back at using either of the keywords; today, yesterday, "this week", "this month", "this year". Note that you need quotes around all keywords consisting of more than one word. In this type of search, the last part of the date or time is "reset" (the time gets set to zero, the day of the month gets set to 1 etc.). The start day of a week is dependent on your server locale. For a en-US locale the first day of the week would be Sunday.  Modified, created and accessed can also be used in comparisons with each other. The example modified::>created locates all items that have been modified after their creation.  Modified can also be used together with a timestamp of ISO 8601 format ("yyyy-MM-dd'T'HH:mm:ssz") to find items modified at a specific time.

Keyword	Example	Function
created	<pre>created:&gt;"this week" created:&lt;"2 weeks ago" created:&gt;"2015-02-01T18:27:55CEST"</pre>	<p>It is possible to search for items that have been created during a specified time span, relative to today. See details regarding the allowed time spans under "modified" above.</p> <p>Modified, created and accessed can be used in comparisons with each other.</p> <p>Created can also be used together with a timestamp of ISO 8601 format ("YYYY-MM-dd'T'HH:mm:ssZ") to find items created at a certain time.</p>
accessed	<pre>accessed:&gt;"this month" accessed:&lt;"2 weeks ago" accessed:null accessed:&gt;"2015-02-01T18:27:55CEST"</pre>	<p>It is possible to search for items that have been accessed during a specified time span, relative to today. See details regarding the allowed time spans under "modified" above.</p> <p>Modified, created and accessed can be used in comparisons with each other.</p> <p>Accessed can also be used together with a timestamp of ISO 8601 format ("YYYY-MM-dd'T'HH:mm:ssZ") to find items accessed at a certain time.</p> <p>The example <code>accessed:null</code> finds all items that have never been accessed.</p> <p>The last example finds all items that have been accessed after the first of February 2015.</p>
::>	<code>modified::&gt;created</code>	<p>Used to find items strictly greater than the expression following the operator.</p> <p>For example, finds all items that have been modified after their creation.</p>
::<	<code>accessed::&lt;modified</code>	<p>Used to find items strictly less than the expression following the operator.</p> <p>For example, finds all items that have been modified after they were last accessed.</p>
<code>parent_id: &lt;folder GUID&gt;</code>	<code>parent_id::538bcde4-7212-475f-a348-5bb41ba39c41</code>	Locates all items located in the specified folder.
<code>format_version: &lt;string or null&gt;</code>	<code>format_version:null</code>	Locates all items of a specified format version. For example, all items which have no format version specified can be found.
<code>version_name: &lt;word in name&gt;</code>	<pre>version_name:final versionname:*certified*</pre>	<p>Locates library items whose latest available version has a version name that starts with the specified word (or part of word).</p> <p>Use wildcards (*) to locate items with the word anywhere in the name.</p>
<code>version_created_by: &lt;username&gt;</code>	<pre>version_created_by:admin versioncreatedby:certificationuser</pre>	Locates library items where the latest available version was modified by a certain user.

Keyword	Example	Function
version_comment: <word in comment>	version_comment:Draft versioncomment:*draft*	Locates all items whose latest available version has a version description that starts with the specified word (or part of word).  Use wildcards (*) to locate items with the word anywhere in the comment.

## Analysis files

When searching for analysis files, there are a number of search parameters that may help you locating a specific group of analyses. If you want to locate analysis files only, you can add `type:dxp` to any search expression.

Keyword	Example	Function
AllowWebPlayerResume: <true or false>	AllowWebPlayerResume:true	If true, locates all analysis files that allow personalized views for all web client users.
EmbedAllSourceData: <true or false>	EmbedAllSourceData:true	If true, locates all analysis files that embed all the source data in at least one data table. ('Embedded in analysis' check box selected.)
AllTablesEmbedded: <true or false>	AllTablesEmbedded:true	If true, locates all analysis files that only have embedded data tables.

## Information model elements

If you want to locate information model elements of a specific type only, add `type:column` (or filter, join, procedure, query, folder or datasource) to the search expression.



If you use TIBCO Cloud™ Spotfire®, Information Designer and the information model elements are not available.

Keyword	Example	Function
description	description:sales type:query description:sales	Locates all items containing the specified word in their description.  Locates all information links containing the specified word in their description.
column	column:Sales column::Sales	Locates all items referring to a source column with the specified name.  The source column could be referred to in the conditions or groupings of a column element, a filter condition, a join condition or the join condition of a procedure.

Keyword	Example	Function
table	table:SalesandCost	Locates all items referring to a source table or stored procedure with the specified name.  This could be referred to in the conditions or groupings of a column element, a filter condition, the condition or target tables of a join or in the source procedure or join condition of a procedure.
schema	schema:dbo	Locates all items referring to a source schema with the specified name.  This could be referred to in the conditions or groupings of a column element, a filter condition, the condition or target tables of a join or in the source procedure or join condition of a procedure.
catalog	catalog:Sales	Locates all elements referring to a source catalog with the specified name.  This could be referred to in the conditions or groupings of a column element, a filter condition, the condition or target tables of a join or in the source procedure or join condition of a procedure.
datatype	datatype:integer	Locates all columns of the specified data type (integer, real, string, date, time, datetime, clob or blob).
parameter	parameter:MinSales parameter:*	Locates information links using the specified parameter.
<property_name>: <property_value>	"my.prop":*	Custom properties in any information model element are searchable using the same syntax.  However, note that the property name must be quoted if it contains a '.' delimiter.

### Combinations of keywords

You can combine many of the keywords described above to create more advanced search expressions. For example:

`(not (required_by(type:dxp))) and type:query` - searches for information links that are not used by any analysis file in the library.


`required_by(type::query InformationLinkName)` - shows the elements used by the information link with the name 'InformationLinkName'.

`(not (required_by(type:dxp))) and type:dataconnection` - searches for data connections that are not used by any analysis file in the library.

`type:dxp depends_on(type:dataconnection title:Salesforce)` - searches for analyses that uses a data connection in the library with "Salesforce" in its title.

`format_version:5.5 OR format_version:5.0 OR format_version:4.5` - searches for old analyses that are saved in versions 5.5, 5.0 or 4.5 and should be re-saved in a newer version.

### To search for items in the Files and data flyout:

1. On the [authoring bar](#), click **Files and data** .

2. Click in the search field at the top of the flyout, and start typing the name of the content that you want to search for.
3. As you type, all the files, folders, and other content that match your search string will be listed.



To reset the flyout to the normal view, click the X button in the search field.



In the installed client, when searching in the **Files and data** flyout you can also search external libraries, like the TIBCO Cloud Data Streams or the TIBCO Data Virtualization integration.

### To search for items in the Spotfire Server Library page:

1. In your client, select **File > View library**.
2. On the **Library** page, click in the search field at the top of the page, and start typing the name of the content that you want to search for.
3. As you type, all the files, folders, and other content that match your search string will be listed.



To reset to the normal view, click the X button in the search field.

You can copy the browser path when you have performed a search, and share it with others, to easily provide a library URL including a search expression. You can also add a search to the library URL manually, but then you must use URL encoding, such as adding '%20' to replace all spaces, in the search query.

Example of a search for all files modified by the user 'cert\_user', less than 10 days ago:



```
https://my-server.my-company.com/spotfire/ui/library/browse/{server id}?q=modified_by:cert_user%20modified:>"10%20days%20ago"
```

Example of a path with a search for items modified by the user 'cert\_user' and also has the word 'sales' in the beginning of title:

```
https://my-server.my-company.com/spotfire/ui/library/browse/{server id}?q=modified_by:cert_user%20sales
```

### To search for items in the Library Administration tool:

An on-premises administrator can use the Library Administration tool in Spotfire Analyst to work with items in the library.

1. In Spotfire Analyst, select **Tools > Library administration**.
2. Type the text you want to search for in the search field at the top right corner of the Library Administration tool.
3. Click on the **Search** button. The Library Administration tool will switch to a Search Result view.
4. The items matching your search criteria are shown in the list. To return to the normal folder view, click the **Back to folder** link.

### To search for items in Information Designer:

An on-premises information modeler can search for items in the Information Designer.

1. In Spotfire Analyst, select **Data > Information designer**.
2. Type the text you want to search for in the search field at the top of the **Elements** tree.

3. Click on the search button with a magnifying glass.



Searching for data sources does not include searching for database entities like catalogs, schemas or tables. It is only the database instance itself that can be located via search.

4. The items matching the search result are shown in the list. To return to the normal folder view, click the **Clear Search** link.

### To use search from the API

A developer can include search for public library item types using the [API](#).

## Creating an RSS feed of your Spotfire Library

With an RSS feed, you can display recent activity in your Spotfire Library. You can create a customized RSS feed that shows the latest changes to the library items you are interested in, narrowed down to certain folders, or to specific search criteria.

To create your own RSS feed, use the following syntax:

```
https://<spotfire-server>/spotfire/library/[library-path]?rss
```

#### Example

```
https://myspotfireserver/spotfire/library/analyses-files?rss
```

The path to a specific library folder is optional. If you only specify `http://myspotfireserver/spotfire/library?rss`, the feed will return the 20 most recently modified files in the library. You can also add a `max-results` section if you want to limit the number of results shown, see example below.

To add a search expression to limit the results, use the following syntax:

```
https://<spotfire-server>/spotfire/library/[library-path]?rss&search=[search-expression]
```

#### Examples

```
https://myspotfireserver/spotfire/library/analyses-files?rss&search==created_by::admin
```

```
https://myspotfireserver/spotfire/library?rss&search=created_by::admin
```

```
https://myspotfireserver/spotfire/library?rss&max-results=10&search=title:sales
```

To learn more about how to search in Spotfire, see [Searching the library](#) on page 124.

## Using search expressions in shareable links

You can incorporate a search expression in a spotfire or tibcospotfire link to assist others finding suitable analyses or data.

See [Links to analyses in the library](#) on page 989 for more information about shareable links. The links are a list of keys and value pairs. The key and value are separated using colon, `:`, and each key and value pair are also separated with colons:

```
tibcospotfire:<key1>:<value1>:<key2>:<value2>...<keyN>:<valueN>
```

The following keys and values are allowed:

Search: <search expression> with optional parameters. See [Searching the library](#) on page 124 for more information about possible searches.

OrderBy: Title | Modified | Created | Accessed | ContentSize | Description

MaxResult: <positive integer>

SortDirection: Ascending | Descending

The values should be encoded using the following pattern:

Value	Encoded to
:	\:
"	\'
\	\\

Examples:

```
tibcospotfire:search:*:OrderBy:Modified:SortDirection:Descending:MaxResult:20
```

```
tibcospotfire:search:modified\:<\'3 days ago'\:OrderBy:Modified:SortDirection:Descending
```

## Opening an analysis saved in the library

You can open analyses saved in the library to continue working with them, or to just view analyses created by colleagues.

### Procedure

1. In the library, click **Browse**.



You can also use search to find the analysis you are looking for, either in the library view, or, in the **Files and data** flyout.

2. Navigate to the analysis you want to open.
3. Click the analysis.

### Result

The analysis is opened.

### What to do next

Explore the analysis or, if you have a Business Author license, enable authoring by choosing **Editing** in the upper-right part of the toolbar.

## Limiting the data to load


If a data source contains large amounts of data, it can take a long time to retrieve all data and the application can be perceived as less responsive when working with it. You might also want to restrict some data from certain users. When you are using information links or data connections as data sources it is possible to limit what data to open in different analyses in a number of different ways.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.



The following methods can be used to limit the amount of data (combinations are also possible):

Method	Use when?	Define where?
On-demand data table	<p>When you want the data in your analysis to dynamically change with some predefined condition. For example, when setting up a details visualization dependent on the marking or filtering in another data table.</p> <p>Another example is when you want one information link or data connection to return different data for different analysis files, in which case you could use the on-demand data table as the only data table in the analysis (with a document property as input).</p>	<p>Using the installed client, on-demand data tables can be added to your analysis by selecting an information link or a data table from a data connection in the <b>Files and data</b> flyout, and then selecting the <b>On-demand</b> option from the drop-down list in the summary view. You must also specify the input conditions that should control loading in the On-Demand Settings dialog.</p> <p>See <a href="#">On-demand data</a> on page 60 for more information.</p> <div>  <p>You can only specify a single fixed value as input to on-demand loading, so if you must retrieve multiple values from a certain column, you have to make sure that an information link is configured to use a multiple selection prompt rather than using it as an on-demand data table.</p> </div>
Prompted and/or parameterized data connections	<p>When the source data amount is huge, but the end users of the data connection are allowed to determine what data to analyze themselves.</p> <p>Parameters defined in the data source can be configured as end user prompts.</p> <p>When you want to keep the data in the external data source (but data connections can also be used to import data).</p>	<p>Using the installed client, prompts can be defined in the Views in Connection dialog or, for SAP BW BEx query connections, in the Data Selection in Connection dialog, as a part of configuring the data connection.</p> <p>Click <b>Define Prompting</b> and specify which column or parameter to prompt for. Mandatory parameters are automatically added as prompts.</p> <p>If the data connection has been created using the installed client and saved in the library, you can add it to analyses using the web client. See <a href="#">Data connections in the Spotfire web client</a> on page 113 for more information.</p>
Details visualizations against external data sources	<p>When you are analyzing in-database data using a connection to an external data source you only load the requested data.</p> <p>By setting up visualizations based on the in-db data as details visualizations limited by the marking or filtering in other visualizations you can make sure that the loaded data is limited to a subsection of the available data only.</p>	<p>Details visualizations can be added using any client but some in-database connections must be configured in the installed client. Make sure that the main data table and the in-db data table have at least one column in common so you get a <a href="#">column match</a>.</p> <p>Right-click on the main visualization and select <b>Create details visualization</b>. Configure the new details visualization to use the in-db data table. See also <a href="#">Drilling down into details</a> on page 597.</p>
Prompted information links	<p>When the source data amount is huge, but the end users of the information link are allowed to determine what data to bring in for analysis themselves. Information links are always analyzed in-memory.</p> <p>Can in some cases be replaced by an on-demand data table.</p>	<p>Using the installed client, prompts can be defined in Information Designer, <a href="#">Information Link tab</a>, <b>Prompts</b> section.</p>

Method	Use when?	Define where?
Personalized information links	When you want the data source to return only information applicable for a certain user name (via a lookup table) or for a specified group or user domain.	Using the installed client, personalized information links can be configured on a filter or column element in Information Designer using the %CURRENT_USER%, %CURRENT_GROUPS% or %CURRENT_USER_DOMAIN% syntax. See <a href="#">Personalized Information Links</a> in the <i>Spotfire Analyst User Guide</i> for more information.
Parameterized information links	When you want the data source to return only information applicable for a certain user or group in a more flexible way than with personalized information links.	Using the installed client, parameters can be created in Information Designer (for example, as a part of an expression set on a column or filter) but their properties and definitions are defined using the API. By using a parameterized information link and a configuration block, it is possible to create an analysis with different input parameters (for example, to be used by an on-demand data table) for different groups of users. See <a href="#">Parameterized Information Links</a> in the <i>Spotfire Analyst User Guide</i> for more information.


# Handling data

Your data is the most important thing when you are working with Spotfire visual analytics. If you are familiar with the data you want to analyze, chances increase that you can set up good visualizations and find interesting conclusions during the analysis.

In some cases, there are things you can do to prepare your data for analysis before you view it in Spotfire, for example, see [Data preparation in Microsoft Excel](#). There are also a number of things you can do within your Spotfire client, to get to know your data, and fix issues before the analysis starts.

## Data in analysis

The **Data in analysis** flyout shows a structured list of all data columns in a data table, that is, it gives you an overview of your data. It can be the starting point for configuring visualizations, because from the flyout, you can drag columns to drop targets in the middle of the visualizations and to the visualization axes. You can also filter to certain values (because each column in the panel is associated with a filter) and you can get recommendations about suitable visualizations and other operations from the expanded flyout.

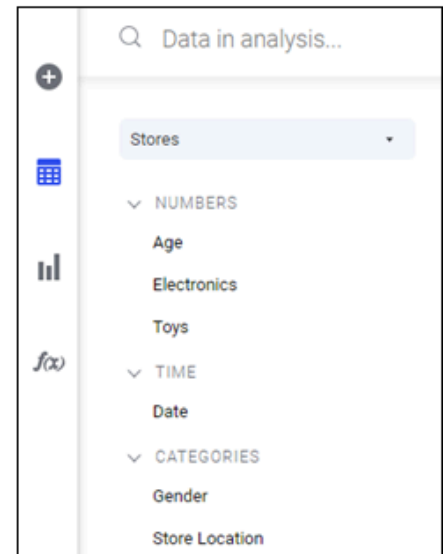
The **Data in analysis** flyout is usually shown when you add new data to an analysis. On the authoring bar, you can also click **Data in analysis**  (when the analysis is in **Editing** mode). If more than one data table is available in the analysis, you can select which data table to investigate from the drop-down list at the top of the flyout.



If you have many data tables you can click on the data table selector and start typing to search for specific words in the data table name.

The columns are by default organized into categories to make them easy to locate. For example, loading the Excel file below results in the **Data in analysis** flyout to the right. Columns containing numerical data suitable to aggregate are gathered in the **Numbers** section, columns related to time are in the **Time** section, and columns useful for splitting data are in the **Categories** section.

	A	B	C	D	E	F
1	Store Location	Age	Gender	Date	Electronics	Toys
2	Los Angeles	61	Female	7/9/2012	10013	11266
3	Seattle	43	Female	7/9/2012	6396	185
4	Boston	55	Female	7/8/2012	5586	354
5	Boston	41	Male	7/5/2012	885	0
6	New York	68	Female	7/1/2012	4860	18918
7	New York	56	Female	6/26/2012	216	304
8	Los Angeles	74	Female	6/24/2012	436	0
9	New York	77	Female	4/18/2012	0	0
10	Los Angeles	62	Male	9/14/2011	153	0
11	Boston	65	Female	6/27/2012	609	1934
12	Boston	54	Female	6/17/2012	1384	91
13	New York	57	Male	4/5/2012	260	963
14	New York	32	Male	7/20/2012	567	45
15	Boston	45	Female	7/12/2012	8332	789
16	Seattle	76	Female	7/5/2012	3334	455
17	Seattle	14	Male	9/14/2011	123	623
18	Los Angeles	16	Male	6/27/2012	0	154



The category of a column affects what happens when you drag the column to a column selector or to a drop target in a visualization. If the suggested category for a column is not the desired category, you can [change the categorization](#).



If you prefer seeing the columns in a sorted list, rather than as categorized columns, you can right-click on the data table selector and select **Sort order** from the pop-up menu. Choose from: Categorized, No sorting, Ascending or Descending views (and, in the installed client, Hierarchical, for cube data).

The **Data in analysis** flyout can be expanded to get more details about a column. It expands automatically when you select columns in the flyout, but you can also expand the flyout yourself by clicking **Expand for details and recommendations** » , at the upper-right side of the flyout.

You can easily change what is shown in a visualization by [dragging a column from the flyout and dropping it onto a column selector](#) or on a drop target within the visualizations. Moreover, you can use the flyout to [filter to certain values](#) in which you are particularly interested.

The sections in the data in analysis flyout differ slightly depending on the data source and the data content. Some examples of differences are described below.

### **In-database relational data from joined tables (connections created using the installed client)**

If data tables from in-db databases have been joined with relations in the Views in Connection dialog (using the installed client), they can be treated as a single, virtual data table within Spotfire. In this case, there will be an additional field showing the **Source structure** in the flyout.

If such a source structure is available, you can click on the different parts of the joined table to see only the columns from each source.

If no relations have been defined, each data table in the data connection will be a separate data table within Spotfire.

### **Cube data (connections created using the installed client)**

When you are working with cube data from Microsoft SQL Server Analysis Services, Oracle Essbase or SAP BW, you will see the underlying structure of the cube in the flyout (as long as you keep the sort order as Hierarchical).

Top levels can be expanded to reveal, for example, the different levels in a hierarchy.

It is recommended to use the hierarchical view when working with cube data, to avoid mixing unrelated columns in one visualization.

See the separate document [Working with Cubes](#) for more information about different cube systems.

## **The expanded data in analysis flyout**


If you click on a column in the **Data in analysis** flyout, the flyout will expand and either show you recommendations or details about the selected column.

If the flyout is not expanded automatically, click **Expand for details and recommendations** » , to the right of the flyout.

The two views of the expanded flyout have different purposes:

- **Details on selected column** gives you an overview of the values in a certain column. When you are in the details view (the column view), you can perform some data preparation and cleansing operations which can help you getting visualizations the way you want them.
- **Recommendations based on selected columns** gives you tips on visualizations or other operations that can help you analyzing the columns selected in the flyout. You can double-click or drag a recommended visualization to the analysis.



To see how the columns in a data table have been built up, on the authoring bar, click **Data canvas** .

#### Example: Details on selected column





The details view for a column will look different for different types of columns. This example shows a string column.

The screenshot displays the 'Data in analysis...' panel in the Spotfire Web Client. The left sidebar contains navigation icons: a plus sign, a table icon, a bar chart icon, a function icon  $f(x)$ , and a lightning bolt icon. The main panel is titled 'Data in analysis...' and shows a dropdown menu for 'Sales Data'. Below this, there are several categories of fields: NUMBERS (Cost, Sales), TIME (Delivery Date, Order Date), LOCATION (Region), CATEGORIES (Buyer, Category, State, Type), and IDENTIFIERS (Number). The 'Region' field is selected and highlighted in blue. On the right side, there are configuration options for the 'Region' column, each with a red numbered callout:

- 1: Region (edit and delete icons)
- 2: No description added (edit icon)
- 3: Unique count (4)
- 4: Categories (South, West, Midwest, Northeast)
- 5: Categorize column as (Location)
- 6: Data type (String)
- 7: Sort order (Standard alphabetical)
- 8: Display values ((Default))
- 9: Geocoding (Not configured)
- 10: Actions (Split column...)

At the bottom, there are tabs for 'ALL' and 'UNIQUE', and a 'Click to sort' button.

1. The column name can be changed by clicking the **Rename column** button , and typing the new name.
2. You can add a description to help others understand the contents of the column and how to use it in the analysis.
3. Depending on the data type and categorization of the column you can see different statistics.  
 If the statistics section shows a histogram, you can click on the histogram in the column view to add a similar visualization to the analysis.
4. You always have the possibility to change the categorization of the column. A good categorization can help you get better recommendations.
5. The data type can always be changed (but for [calculated columns](#) you edit the expression to change the data type rather than using this drop-down list).
6. You might also have the opportunity to change things like the sort order or formatting of the column, or, you might be able to handle empty values in the column. The preferred aggregation method (for numerical data) can be specified to improve recommendations and make it faster to work with the column in visualizations. Which options are available depends on the column type.
7. Display values allows you to show values from a different column or expression on categorical axes and in filters. See [Setting display values based on other columns](#) on page 159 for more information.
8. Some columns can be used for placing your data on maps. Read more in [Geographic location and geocoding](#) on page 169.
9. You might also have access to additional actions that are specific for the data type of the column, such as the [Split column](#) action that can be used to split string columns.
10. At the bottom of the column view you see the contents of the column; either all values or unique (distinct) values only.

### Example: Recommendations based on selected columns



If you select multiple columns in the flyout, you will get recommendations on how you can visualize those columns together:

**Recommended Visualizations:**

- Create bar chart:** View Sales per Category and Type. (Visualized as a stacked bar chart with three bars of different colors).
- Create treemap:** View Sales per Category » Type. (Visualized as a treemap showing hierarchical data for Fruit, Spices, and Vegetables).
- Create cross table:** Show Sales per Type and Category. (Visualized as a table with categories and types).

Category	Apples	Bananas	Cinnamon
Fruit	465952.48 kr	718291.43999...	
Spices			294457.58

Click or drag a recommended visualization to the page to add it to the analysis.

You might also get other types of recommendations here, such as the option to link two similar data tables to each other (add a relation), so that markings in visualizations based on one of the data tables also affect visualizations based on the other data table.

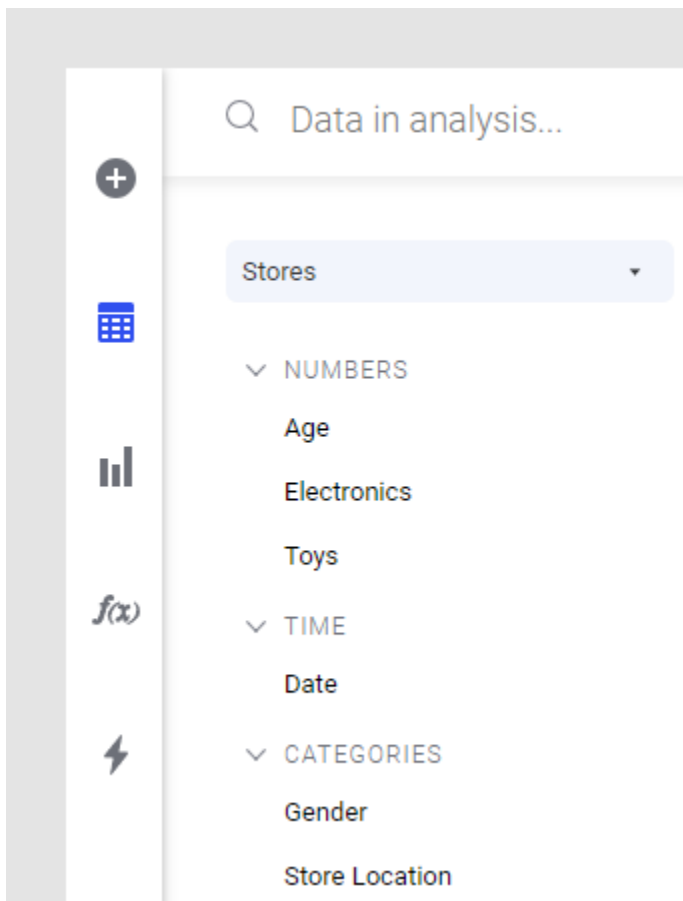
## Changing the categorization of a column

In the **Data in analysis** flyout, the columns are by default organized into categories. The category of a column affects what happens when you drag the column and drop it on a column selector or drop target in a visualization. It also affects which kinds of visualizations are advised in recommendations. If the suggested category for a column is not your preferred category, you can change the categorization.

### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.





The different categories are Numbers, Currency, Time, Location, Categories, Identifiers, Text, Images, and Binary. Which categories are available depends on the data type of the column. For example, if the values in a column are integers like 4633, 4637, and 4638, the suggested category might be Numbers. This means that you can calculate sums or averages, or use other aggregation methods. However, these numbers could just as well represent, for example, employee numbers or purchase order numbers, and sum or average would not be relevant. In these cases, the column values should instead be handled as Identifiers or as Categories.

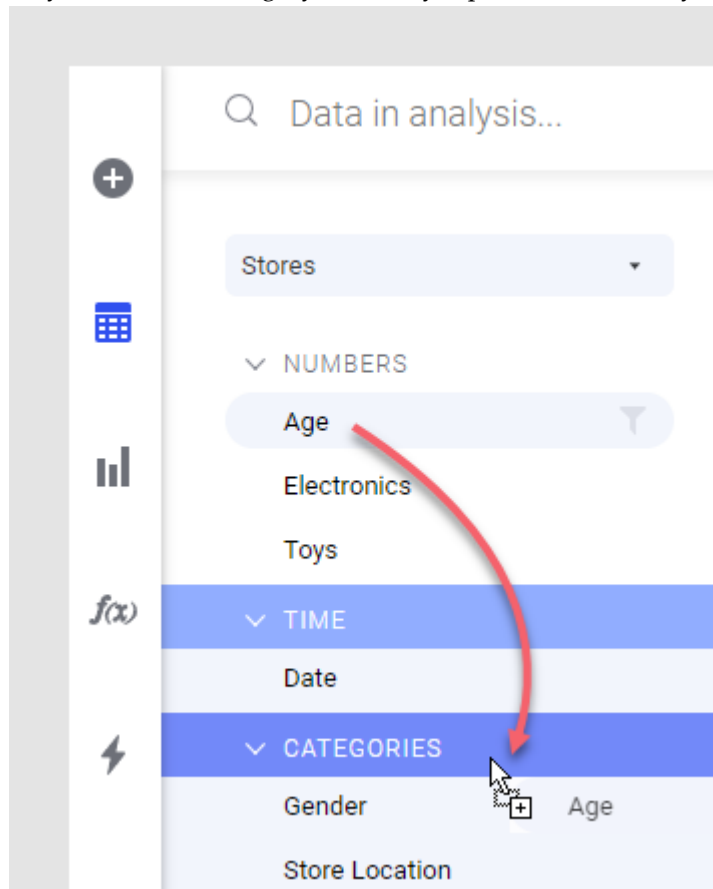
### Procedure

1. In the **Data in analysis** flyout, right-click on the column whose categorization you want to change.  
You can also expand the **Data in analysis** flyout and change the categorization for a certain column in the **Categorize column as** section of the column view.
2. In the opened menu, select **Change categorization**.

3. In the list of categories, select the preferred category.

Only relevant categories can be selected. If a desired category is unavailable you might need to [change the data type of the column](#) first.

- Alternatively, drag the column to the header of the preferred category. However, this alternative only works if the category is already represented in the flyout.



### Result

The column is moved to the selected category.



You can change the category for several columns simultaneously. Select the columns in question in the **Data in analysis** flyout (by selecting their check boxes), and they will all be included in the move to the new category using the right-click or drag-and-drop methods.



Right-clicking a column and selecting **Change categorization** offers the **Reset to default** option. If you select this option, the categorization of the column will be returned to the original suggestion (based on data import heuristics).

## Changing the formatting of a column

The formatting of a column defines how the data in the column will be presented. For example, depending on how you format a date column, the dates therein can be shown as "01/12/2022" or "January 12th, 2022". When you define formatting at the column level (in the **Data in analysis** flyout), this is the default formatting that will be used at all places where the column content is shown.

### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.






You can also specify formatting on an axis in a certain visualization using the visualization **Properties** or using short-cuts accessed by clicking or right-clicking on the axis. In those cases, the formatting changes only apply to this particular axis. The types of formatting available are the same as when changing the formatting on the column.



In the **Data in analysis** flyout, you can also right-click the column of interest and select **Formatting** from the pop-up menu.

### Procedure

1. On the **authoring bar**, click **Data in analysis**  and click on the column of interest.
2. In the expanded flyout, make sure that **Details on selected column**  is selected.
3. In the expanded flyout, locate the **Formatting** section and either change the formatting to a different type directly from the drop-down list or click **Settings** , and make further modifications in the Formatting settings dialog as described below.
4. In the Formatting settings dialog, click the tab of interest.  
Depending on the data type, you will have access to different tabs and settings in the dialog. Numeric columns have more options to choose from than other types of columns. The **Custom** format allows you to specify your own format, if what you are looking for is unavailable in the predefined formats.
5. Select the formatting you want to use from the controls in the dialog, or, define a custom formatting. Alternatively, you can also **Copy formatting settings from another column** in the analysis, by expanding the section at the bottom of the dialog and choosing a column to copy formatting from. A sample of the selected format is automatically shown in the dialog when you apply an update.
6. When you are done, click **OK**.

### Result

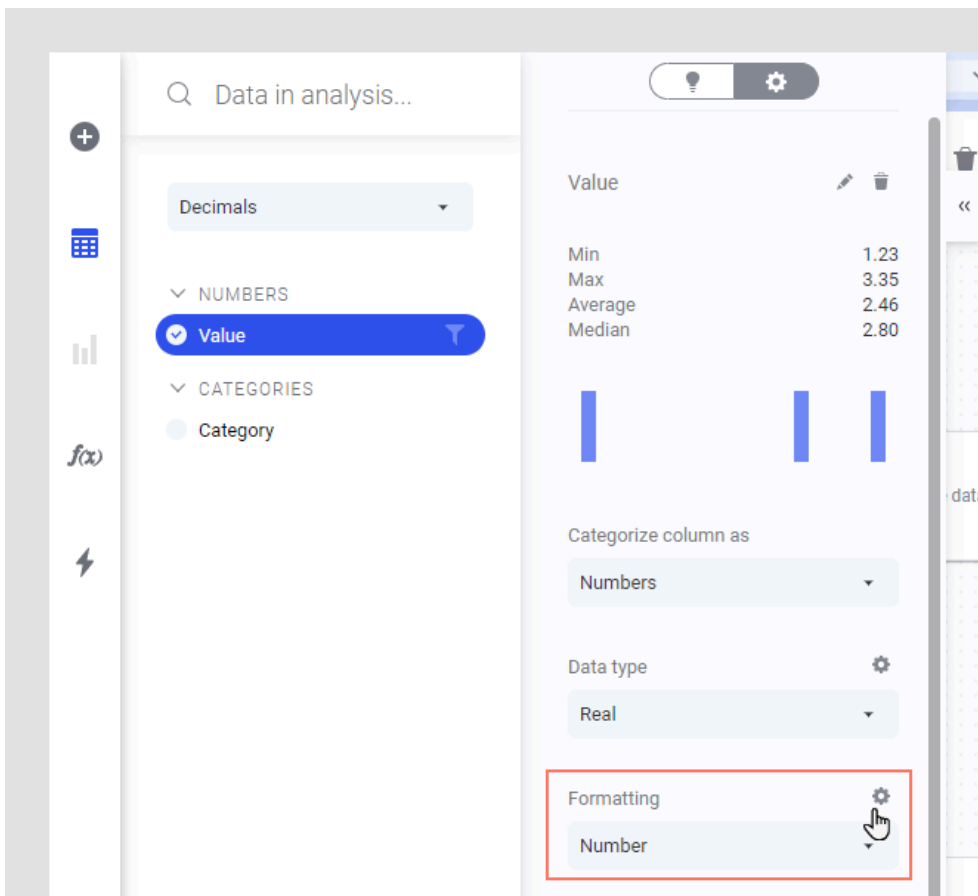
When you have defined formatting at the column level, this is the default formatting that will be used at all places where the column content is shown.

### Example: Changing the number of decimals

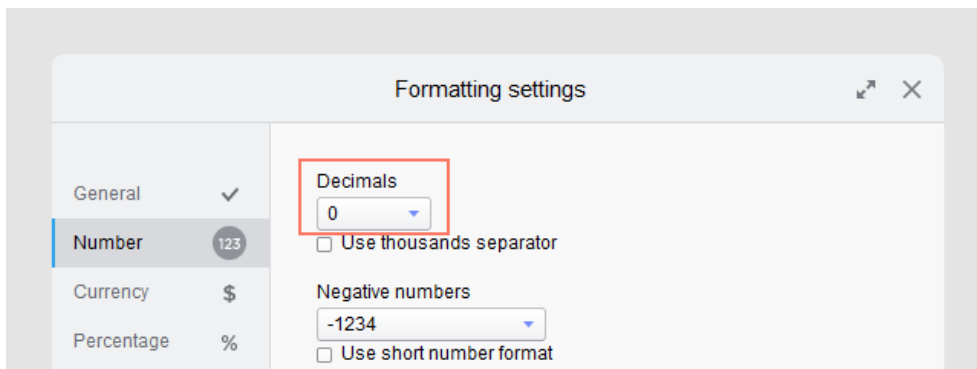
If your data contains measures with many decimals, like the values below, but you are interested in displaying integers only, you can change the default formatting so no decimals are shown.

Category	Value
A	2.795346
B	3.346743
C	1.234583

Show the Formatting settings dialog from the expanded **Data in analysis** flyout by clicking **Settings** ⚙️.



Then change the number of decimals to zero.



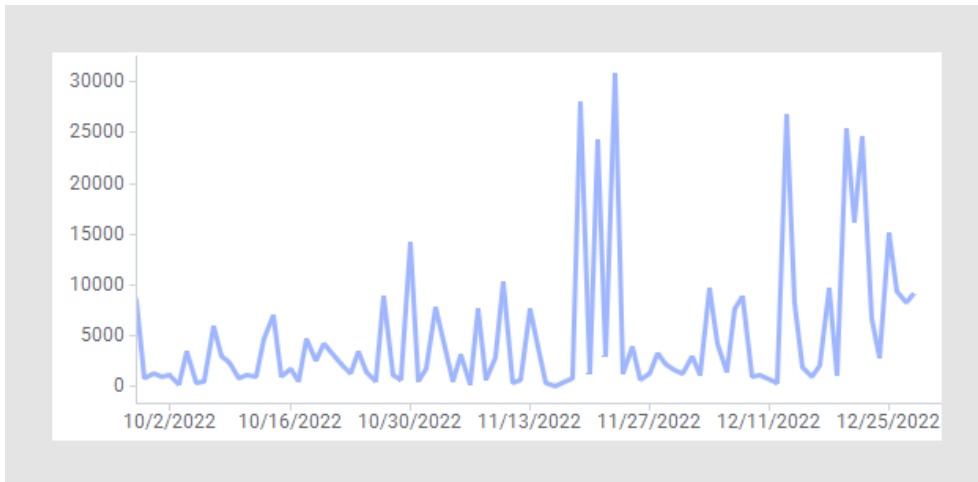
The visualization now shows the values without any decimals:

Category	Value
A	3
B	3
C	1

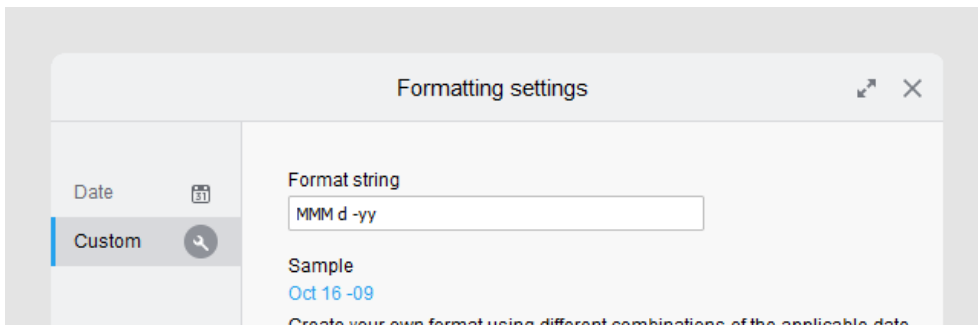
### Example: Defining a custom format for a date column

With custom formatting you can specify exactly how you want a value to be presented. This example shows how you can define a custom format for a date column.

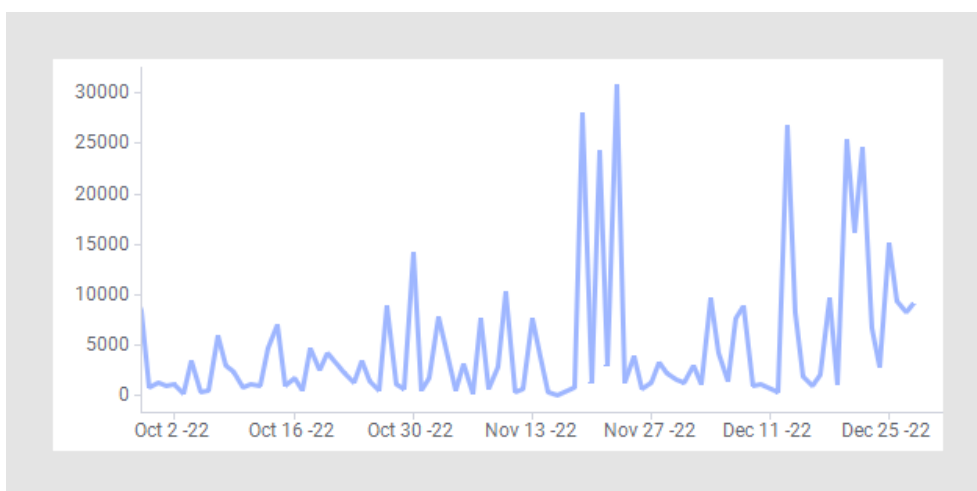
Which predefined formats are available for a date or time column depends on the language setting on your computer.



If you want to define your own format, in the Formatting settings dialog, click **Custom** and type the format string that specifies your format. In this example, a short format of the month name is shown, together with the day, expressed as a number, and short year: MMM d -yy. See [Custom formatting of date and time columns](#) for more details about custom formats.




Click **OK** and see the result in the visualization:



## Custom formatting of numeric columns

If the format you want to use cannot be created with the given settings, the custom format string allows you to create your own formats using a code explained in the examples below.

The special characters explained below allow you to multiply, divide, separate numbers, etc. Any other characters included in the format string are printed in the resulting data.

Character	Description
0	Always returns a value for the position it is written in. If there is no number in its place in the data, 0 (zero) will be used.
#	<p>Returns values if there are numbers in its place in the data.</p> <p>If used to the left of the decimal point, all digits are returned even if there is one # in the format string and three digits in the data.</p> <p>If used to the right of the decimal point, the same number of digits are returned as there are # to the right of the decimal point, and the number gets rounded up or down. See example below.</p>
,	<p>If used before a decimal point, it divides the number in the data by 1000.</p> <p>A difference from Excel is that Excel allows for " , " as divider after the decimal point as well.</p>
%	Multiplies the number by 100 and inserts a "%" in the number, at the location where it is written in the format string.
.	<p>Decimal point.</p> <div>  <p>If no decimal point is used and there are decimals in the value you apply the format string on, the value gets rounded up or down.</p> </div>
;	<p>Used to divide a format string if different formats are to be used for positive numbers, negative numbers and 0 (zero).</p> <p>If no semicolon is used, the format string is used for all numbers.</p> <p>If one semicolon is used, it divides the format string like this: String for positive numbers and zero;String for negative numbers</p> <p>If two semicolons are used, they divide the format string like this: String for positive numbers;String for negative numbers;String for zero</p>

Character	Description
\	If a "\" is added before a special character that character will not modify the number, the character will only be added to the value.

### Examples:



All these examples use the number 12345.67 as the value from the data.

Format string	Result
# ####	1 2346
#.#	12345.7
#.000	12345.670
#,.#	12.3
#,,.##	.01
#%	1234567%
#\%	12345.67%
\$#	\$12346
#.##E+0	1.23E+4
#.#;( #.#)	12345.7 Had the number been negative, the result would be: (12345.7)
23	23

For more information, see literature about custom numeric format strings (for example, on MSDN).

### Custom formatting of date and time columns

If the format you want to use cannot be created with the given settings, the custom format string allows you to create your own formats using a code explained in the examples below.

### Examples:

Below are some examples of custom format strings for datetime formats. For more information, see literature about custom `DateTime` format strings, such as that on MSDN.





If you want to use any of the custom date and time format specifiers alone in a format string (for example, to use the "d", "h", or "M" specifier by itself), you must either add a space before or after the specifier, or include a percent sign ("%") before the single custom date and time specifier, to avoid it being interpreted as a standard format string.

Character	Description
yy	Returns the year, measured as a number between 0 and 99.
yyyy	Returns the year as a four-digit number.
M	Returns the month, measured as a number between 1 and 12, with one or two digits depending on the value.  This means that June will be written as '6', when this format string is applied (whereas November is written as '11').
MM	Returns the month with two digits, measured as a number between 1 and 12. This means that June will be written as '06'.
MMM	Returns the abbreviated name of the month. For example, 'Jun'.
MMMM	Returns the full name of the month. For example, 'June'.
d	Returns the day of the month, measured as a number between 1 and 31, with one or two digits depending on the value.  This means that the 6th of a month will be written as '6' (whereas the 11th is written as '11'), when this format string is applied.
dd	Returns the day of the month with two digits, measured as a number between 1 and 31. This means that the 6th of a month will be written as '06'.
ddd	Returns the abbreviated name of the day of the week. For example, 'Fri'.
dddd	Returns the full name of the day of the week. For example, 'Friday'.
h	Returns the hour using a 12-hour clock, with one or two digits depending on the value.  This means that if the time is 7:20:22, then the format 'h' will display the hour as '7', that is, using a single digit. If the time is 11:20:22, then 'h' returns 11.
hh	Returns the hour using a 12-hour clock which always uses two digits.  This means that the time 7:20:22 will be written as '07:20:22' when the 'hh' format string is applied.
H	Returns the hour using a 24-hour clock, with one or two digits depending on the value.  This means that if the time is 7:20:22, then the format 'H' will display the hour as '7', that is, using a single digit. If the time is 11:20:22, then 'H' returns 11. If the time is 20 minutes past seven in the evening, then 'H' returns 19:20:00.
HH	Returns the hour using a 24-hour clock which always uses two digits. This means that 6 o'clock in the morning will be written as '06' and 6 o'clock in the evening will be written as '18'.
m	Returns the minute with one or two digits, depending on the value. This means that six minutes will be written as '6' (whereas 11 minutes are written as '11').
mm	Returns the minute with two digits. This means that six minutes will be written as '06'.
s	Returns the second with one or two digits, depending on the value.  This means that six seconds will be written as '6' (whereas 11 seconds are written as '11').

Character	Description
ss	Returns the second with two digits. This means that six seconds will be written as '06'.
f	Returns the tenths of a second.
ff	Returns the hundredths of a second.
fff	Returns the milliseconds.
tt	Returns the AM/PM designator.
:	Returns the time separator.
/	Returns the date separator.

You can also add any custom string value, but if any of the specifier characters are included in the string, they need to be escaped by a backslash (\).



All the examples below use the following value from the data: Friday, October 16, 2009, at 25 minutes past three in the afternoon.

Format string	Result
dd\t\h o\f MMMM yyyy	16th of October 2009
MMM d yyyy, HH:mm	Oct 16 2009, 15:25
\year: YY, \mon\t\h: MM, \da\y: dd	year: 09, month: 10, day: 16
hh:mm tt	03:25 PM
m \minu\te\s pa\s\t h, MMM d	25 minutes past 3, Oct 16
%d	16

## Custom formatting of timespan columns

If the format you want to use cannot be created with the given settings, the custom format string allows you to create your own formats using a code explained in the examples below. There are five different data values included in the `TimeSpan` format: days, hours, minutes, seconds and fractions of seconds.

### Examples:

These can be combined to a suitable format using a format string built by the following specifier characters:

Character	Description
d	Returns the number of days.
h	Returns the number of hours with one or two digits, depending on the value. This means that six hours will be written as '6' (whereas 11 hours are written as '11') when this format string is applied.
hh	Returns the number of hours with two digits. This means that six hours will be written as '06', when this format string is applied.

Character	Description
m	Returns the number of minutes with one or two digits, depending on the value. This means that six minutes will be written as '6' (whereas 11 minutes are written as '11') when this format string is applied.
mm	Returns the number of minutes with two digits. This means that six minutes will be written as '06'.
s	Returns the number of seconds with one or two digits, depending on the value. This means that six seconds will be written as '6' (whereas 11 seconds are written as '11') when this format string is applied.
ss	Returns the number of seconds with two digits. This means that six seconds will be written as '06'.
f	Returns the fractions of seconds. When specifying custom format strings, you can also add a number between 1 and 3 after the f, defining how many decimals are shown. If no number has been specified, three numbers are shown, if available.

Between each specifier character, you need to supply some kind of separator. This could be a custom string value, but if any of the specifier characters are included in the string, they need to be escaped by a backslash (\). You can also include an initial and a conclusive string.

All the examples below use the following value from the data: -5 days, 7 hours, 11 minutes, 3.1234 seconds.

Format string	Result
d.h:m:s.f	-5.7:11:3.1
d.hh:mm:ss.f2	-5.07:11:03.12
Ti\me\span i\s d \day\s	Timespan is -5 days
d \day\s h \hour\s m \minute\s s \secon\d\s	-5 days 7 hours 11 minutes 3 seconds
f s m h d	123 3 11 7 -5

## Formatting details

Formatting is used for things like specifying the number of decimals to show, to determine how to present dates or times, or to give a value meaning by adding units of measurements, thousands and decimal separators, and so on. Locale settings, determined by Windows Regional Settings, are often used to determine the formatting, but you can also make certain changes yourself.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.



Formatting does not include visual properties, such as color, font, or size.

Example: If you are running the installed client with data values listed in US currency, you have Windows Regional Settings set to Swedish and you select to show thousands separators, you will get formatted values such as \$1.000.000,00. If you change your locale settings to US English, the value will be re-formatted into \$1,000,000.00.

The available formatting options depend on the data type of the value. Text cannot be formatted at all, while an integer has several different possibilities. An integer can, among other things, be formatted as a number, a currency, and as a percentage. You can specify the number of decimals to show, or whether

to use a thousands separator. Another possibility is to use short number format, which is a way to shorten values to take up less space by replacing powers of tens with symbols. See [Defining a new short number format](#) on page 155 to learn more. You can apply formatting to your data on different levels and you can access the settings in different ways as described below.

### Default axis formatting

In the installed client, you can use the Options dialog to specify default formatting for different data types on the visualization axes. For example, if you specify the data type integer to be formatted as US currency (\$) with two decimals, then each time you create a new visualization, values on an axis with integers will be shown as US currency with two decimals. Note that these default formatting settings do not affect the columns or the data tables in any way, only the axes in the visualizations.

To reach the default settings, select **Tools > Options** on the menu bar, and go to the **Axis formatting** page in the dialog.

### Column formatting

If you want a specific column in a data table to be formatted in a certain way throughout the analysis, you can format values on a column level. For example, if you format a column called "Cost" as US currency (\$) with two decimals, the values in the column "Cost" will be displayed as US currency with two decimals every time that column is used. These settings affect all the places where the column is used in the analysis, including axes in visualizations, filters, tooltips, etc., and changes take effect immediately. Column formatting overrides formatting on default axis formatting level.

To change formatting settings on column level, see [Changing the formatting of a column](#) on page 142.

### Formatting in visualizations


You can also format values in individual visualizations. For example, if you format the values on the X-axis of a scatter plot as Number with no decimals, the values on the X-axis in that scatter plot will be instantly updated with the new formatting settings. The Y-axis in the scatter plot will not be affected by the new settings, and no other visualizations in the analysis will adopt the changes. Formatting on this level overrides settings on both the default axis level and on the column level. You can define formatting settings on the axes in many of the visualizations, but not all. Formatting on visualization level is similar in all kinds of visualizations, but the cross table differs slightly and is described in [Formatting in the Cross Table](#) in the *Spotfire Analyst User Guide*.

To change formatting settings on the visualization level, use the Visualization properties for the axis of interest.

If you use more than one scale in a visualization, you can format each of the scales individually, the same way you format an axis in a visualization. However, if there is more than one scale on the same side of an axis in a visualization, you cannot format the scales on that side separately by right-clicking directly on the scale. This will instead change the settings for all scales. However, you can still format the scales separately from the Visualization properties.

No matter on which level you change the format, the options are the same. See the table below to get a full description of all possible options. Some of the options listed are subcategories of other options.

Option	Description
Boolean	A format used to show boolean data.

Option	Description
Currency	<p>A number format for currencies. A currency culture can be other than the current culture. In that case, number formatting is done according to the current culture, but the currency symbol and pattern is defined by the specified currency culture.</p> <p>Examples:</p> <p>\$ 1234.45</p> <p>¥1,234</p> <p>-1,234.45 kr</p> <p>(\$ 1,234.45)</p>
Custom	<p>A highly configurable format, which supports custom format specifiers.</p> <p>Examples:</p> <p>(1.234E+099)</p> <p>999,999.00 Dollars</p> <p>"d" Short date pattern</p> <p>"T" Long time pattern</p> <p>"G" General date/time pattern</p> <p>Read more about custom formats for <a href="#">numeric columns</a>, <a href="#">date and time columns</a> and <a href="#">timespan</a> columns on their respective pages.</p>
Date	A format used to show dates.
DatePart	Specifies that the column contains a date part and allows you to show the date part value as a string instead of a number, where applicable.
Decimals	<p>Specifies the number of decimals (0-15) to show. An additional alternative (Auto) is also available. If you select (Auto), the number of decimals will be different from one number to another since trailing zeros will not be included.</p> <p>Examples:</p> <p>1,234.560700 (six decimals)</p> <p>1,234.560 (three decimals)</p> <p>1,234.56 (two decimals)</p> <p>1,234.5607 (Auto)</p>
General	<p>A general, non-configurable format, with a variable number of decimals and no thousands separator. The number is converted to the most compact of either fixed-point or scientific notation. A minus sign is always used for negative values.</p> <p>Examples:</p> <p>1</p> <p>12345</p> <p>1.23456E-6</p> <p>-12345</p>
Negative numbers	<p>Specifies the pattern used for showing negative numbers [-1.23 or (1.23)].</p> <div>  <p>The negative number format setting in Windows Regional Settings is ignored.</p> </div>

Option	Description
Number	<p>A configurable number format.</p> <p>Examples:</p> <p>1234.45</p> <p>1,234.45</p> <p>-1,234.45</p> <p>(1,234.45)</p>
Percentage	<p>A percentage number format, where percentage symbol and format pattern is culture specific. The converted number is multiplied by 100 to be presented as percentage. A minus sign is always used for negative values.</p> <p>Examples:</p> <p>100 %</p> <p>- 99 %</p>
Scientific	<p>A number format for scientific notation with a fixed number of decimals. A minus sign is always used for negative values.</p> <p>Examples:</p> <p>1.00000E+099</p> <p>5.6E-001</p> <p>-5.6E-001</p>
Time	A format used to show times.
TimeSpan	A format used to show time spans.
Type	<p>Specifies the form on which values should be shown from a predefined list. What you see in the list depends on the data type and the locale.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>• Short date: 10/16/2023</li> <li>• Long date: Monday, October 16, 2023</li> <li>• Short time: 3:25 PM</li> <li>• Long time: 3:25:55 PM</li> <li>• Short date/short time: 10/16/2023 3:25 PM</li> <li>• Long date/long time: Monday, October 16, 2023 3:25:55 PM</li> <li>• Month day: October 16</li> <li>• RFC1123 : Fri, 16 Oct 2009 15:25:55 GMT</li> <li>• Sortable (ISO 8601): 2009-10-16T15:25:55</li> <li>• Year month: October, 2009</li> <li>• 25 days 09 hours 03 minutes 07 seconds 015 milliseconds</li> <li>• 25.9:3:7:015</li> <li>• True/False</li> <li>• 1/0</li> </ul>

Option	Description
Use engineering notation	Use this option to write the scientific number format with the exponent of 10 always expressed as a multiple of three. Examples: -12.35E+003 12.35E+006
Use short number format	Use this option to shorten numerical values, to take up less space. For example, 1,000 can be formatted to 1k. Select which symbol set to use from the drop-down list. To learn more about symbol sets, see <a href="#">Defining a new short number format</a> on page 155.
Use thousands separator	Specifies whether to use a thousands separator (digit grouping symbol in Windows Regional Settings) or not. The grouping symbol cannot be explicitly specified (only through Windows Regional Settings).

## Defining a new short number format

If the values on an axis or a column are numerical, you can choose to display them in short number format. This means that values with many digits can be shortened to take up less space. For example, by using short number format you can set the number 1,000 to be shown as 1k. A standard symbol set is already defined and available to use. The defined symbols in the standard symbol set are M for  $10^6$ , and k for  $10^3$ . You cannot delete or change the standard symbol set. However, you can add your own symbol sets and define symbols of your choice in the installed client.

### Prerequisites

New short number formats must be added using the installed client.

As with other formatting settings, you can apply short number formatting in different ways and on different levels. See [Changing the formatting of a column](#) on page 142 for more information. For specific details about formatting settings, see [Formatting details](#) on page 151.

### Procedure

1. In the installed client, open **Tools > Options** and go to the **Axis formatting** page.
2. Click **Edit Symbol Sets**.
3. Click **Add** in the upper part of the dialog.
4. Type a name to use for the new symbol set in the text field and click **OK**.  
The symbol set is added to the drop-down list.
5. Click **Add** next to the **Defined symbols** list to add symbols to the symbol set.  
You should add all abbreviations that should be available to the values in a column to your new symbol set. For example, you could add k, M, G, T, if you expect to have a wide range of large numbers in a column.
6. Enter an exponent and a symbol to replace it with and click **OK**. Repeat for the next symbol.
7. Click **OK** to close all dialogs.

### Result

The new symbol set is added to the list of defined symbols under **Use short number format** in the Formatting settings dialogs.

## Changing the default sort order of values in a column

Text-based columns (strings) can be sorted differently depending on whether you want digits in the texts to be treated as numbers or to be sorted strictly alphabetically. You might also want to define a custom sort order, if the strings in the column have a hierarchical structure that is not alphabetical.

### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.






You can only change the sort order for text-based columns.



In the flyout, you can also right-click the column of interest and select **Sort order** from the pop-up menu.

### Procedure

1. On the [authoring bar](#), click **Data in analysis**  and click on the column of interest.
2. In the expanded flyout, make sure that **Details on selected column**  is selected.
3. In the expanded flyout, locate the **Sort order** section and either change the sort order directly from the drop-down list or click **Settings**  and make further modifications in the Sort order dialog as described below.
4. In the Sort order dialog, click the sort order type of interest to the left.  
A sample of the selected sort order is shown in the list to the right. If you select a **Custom** sort order you can define the custom sort order to use by moving values up or down in the field to the right.  
Alternatively, you can also **Copy sort order settings from another column** in the analysis, by expanding the section at the bottom of the dialog and choosing a column to copy the sort order from.  
See below for examples of sort order settings.
5. When you are done, click **OK**.

### Result

When you have defined a new sort order for a column, this is the default sort order that will be used at all places where the column content is shown.

#### Example: Standard alphabetical sort order

By default, all text-based columns will be sorted alphabetically. This means that digits and non-alphanumeric characters will be sorted according to their position in the alphabet. The following values will be sorted as seen below:

```
a#
a1
a20
a3
b1
```

The number 2 comes before the number 3 in the alphabet, so a20 comes before a3 in the list.



### Example: Alphabetical (natural sort of numbers)

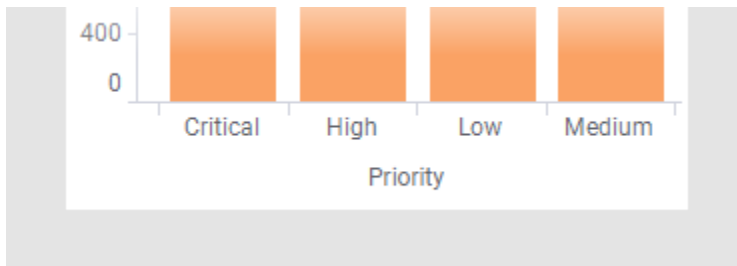
If the digits included in the texts should be treated as numbers, you would instead use the alphabetical sorting with natural sort of numbers:

```
a1
a3
a20
a#
b1
```

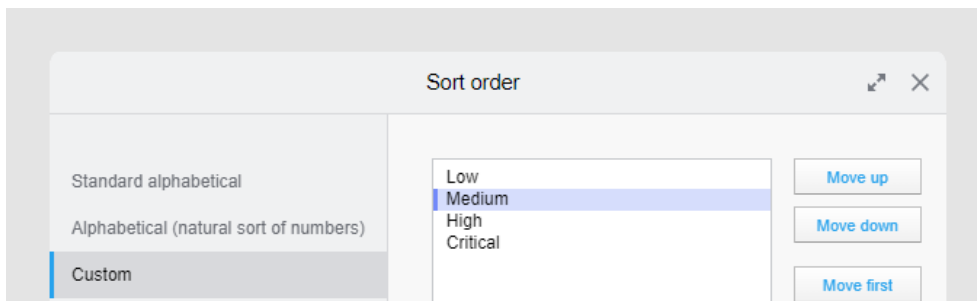
Then, the value a20 will be seen as a higher number than a3, as shown above.

### Example: Custom sort order for column with priorities

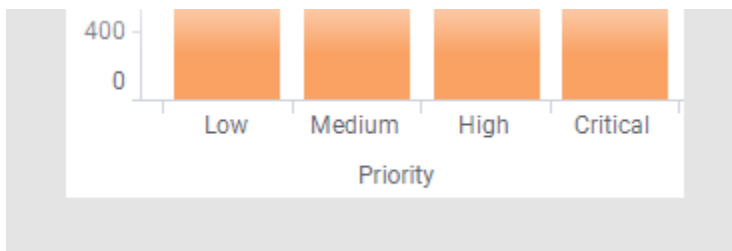
If your data contains values that you want to display in a certain order, you can define a custom sort order for that column. For example, with a column called 'Priority', containing four different values, the default sort order would give the following result:



If you instead want the bars to be in an order from **Low** to **Critical**, you can set up a custom sort order for the **Priority** column by moving the values up or down in the Custom sort order dialog using either the **Move up**/**Move down** buttons or drag-and-drop:



The resulting order in the visualization will then be as follows:






## Changing the data type of a column

The data type determines the possible formats for the values in a column and sometimes it also limits which axes a column can be placed on, and the operations that can be done with the column. If the data type is wrong, you may be unable to use your data the way you want. It is easy to change the data type in the expanded **Data in analysis** flyout.

### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.

### Procedure

1. On the [authoring bar](#), click **Data in analysis**  and click on the column of interest.
2. In the expanded flyout, make sure that **Details on selected column**  is selected.
3. In the expanded flyout, locate the **Data type** section and change the data type.
4. In some cases, you may also need to click **Settings** , to define how the values in the column should be converted to the new data type.  
Different conversions require different settings. For example, if you want to convert a `String` into a `Date` you may need to specify the locale used in the string, or, define how the date should be interpreted using a custom format string.

### Result

The data type is immediately updated, both in the data flyout and in any visualizations.

## Changing a column name

The column name is shown in table headers and on the axes of other visualizations. With a proper column name it is easier to understand what a visualization shows. Authors can rename columns in the **Data in analysis** flyout, and also add or update the column description.

### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.





You can also right-click on the column in the **Data in analysis** flyout and select **Rename** from the pop-up menu.

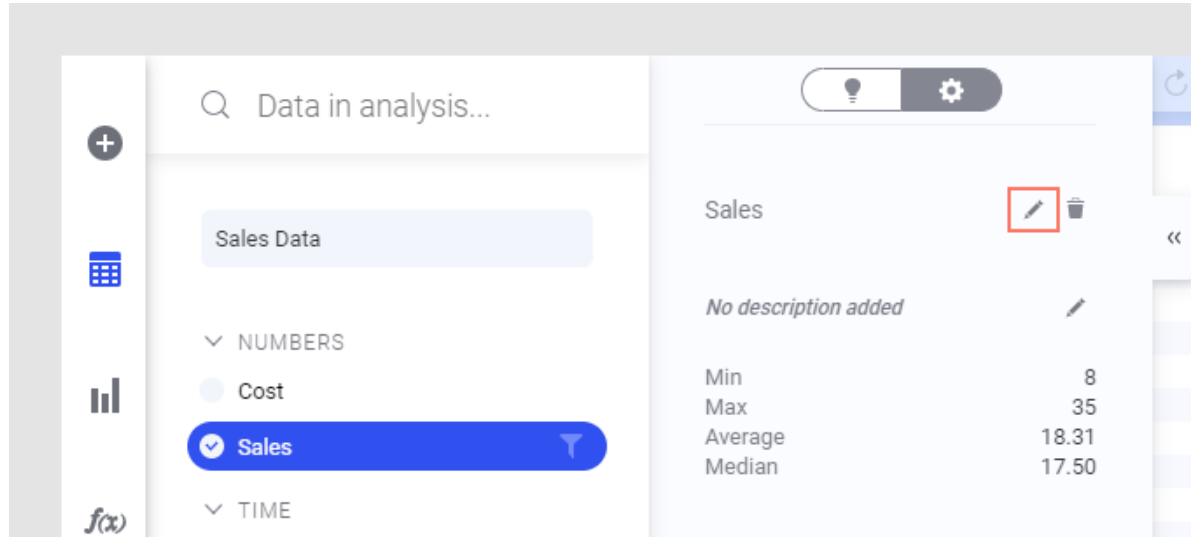


If you only want a different name on an axis in a specific visualization, you can [change the display name](#) instead.

### Procedure

1. On the [authoring bar](#), click **Data in analysis**  and click on the column of interest.
2. In the expanded flyout, make sure that **Details on selected column**  is selected.

3. In the expanded flyout, click **Rename column**.



You can also double-click on the column name in the expanded flyout to enable editing.



Similarly, you can add or edit the description of the column by clicking the pen icon, or by double-clicking the text field for the description.

4. Type a new name for the column and finish by pressing Enter.  
The column name is updated in all visualizations.



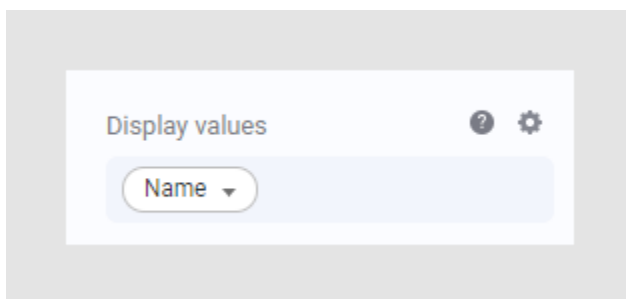
The columns use the same name as their corresponding filters, so renaming a column will also change the filter name (and the other way around), if a filter exists for the column.

## Setting display values based on other columns

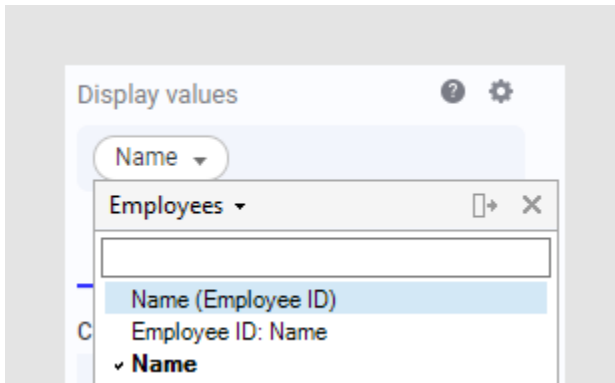
Sometimes, you might want to show other values than the ones that actually define an axis as the display values. Display values can easily be configured in the expanded **Data in analysis** flyout.

For example, you might have data for customer IDs in an external data table (e.g., streaming data), together with lots of other information, but the mapping of the IDs to actual customer names is included in a smaller, imported data table. When one of the data tables is in-database it is not possible to add columns directly to join the two tables. Then, you can choose to show the data based on the external IDs in a visualization, but to change the display values to show the customer names from the imported data table as the axis labels instead. (Data and display values alike can come from either in-memory or in-db data tables when using display values.)

Display values are configured in the expanded **Data in analysis** flyout. Only string columns or expressions are allowed as display values.



When defining display values and there is a single matching string column available in the target data table, you can choose to show just the value from the other column, or a combination of the old value and the display value from the other column:



It is possible to specify similar configurations even if the shortcut options are unavailable. Do it by defining a custom expression on the display value column (such as, `Concatenate([Employees].[Employee ID], " : ", [Employees].[Name])`). Note that expressions that mix columns from multiple data tables are not supported.





Currently, only columns classified as identifiers or categories can have display values. If you select display values for another type of column then it will automatically be re-categorized as 'Categories'. This makes sure that expressions using this column expect a categorical expression instead of a continuous one. You cannot use display values with cube data and it is not possible to use images or geometry data as display values.

To use column values from a different data table, there must be a column match between the two data tables. This is often created automatically (for example, if the name and data type of two ID columns in the data tables are the same) but it is also possible to configure specific column matches using Spotfire Analyst. See [Multiple data tables in one visualization](#) on page 206 for general information about matching data tables.



When changing the display values for a column, the display values will be shown on all places where they make sense to show up, for example, as axis labels on categorical axes, in tooltips in visualizations, or in check box-, radio button-, item- or list box filters. Range filters and other continuous applications will always show the original values. Searching in text filters is always performed against the original data, so search for a display value will not result in any hits, whereas searching in list box filters will show hits for both display values and original values.

## Procedure

1. On the [authoring bar](#), click **Data in analysis**  and click on the column of interest.  
This should be the column you want to change the display values for.
2. In the expanded flyout, make sure that **Details on selected column**  is selected.
3. In the lower part of the expanded flyout, click the column selector under **Display values**.  
If there is more than one data table in the analysis, and a column match is available, you will see a drop-down list at the top of the expanded column selector that lets you choose a data table. If you have a single data table in the analysis you can only choose columns from that data table. See [Choosing the data to analyze](#) and [Choosing how to load the data](#) for help on how to add another data table to the analysis.

4. In the upper part of the column selector, choose the data table from which to pick the values for the display values.

If you cannot see any columns in the list, you might not have a column match defined. See [Column matches](#) on page 212.

5. Click on the column to provide the display values.

You can also define a custom expression for the display values. For example,

`Concatenate([Employees].[Firstname], " ", [Employees].[Last name])`. See below for more details.

Visualizations and applicable filters are updated to show the display values instead of the original values.



To exclude an axis in a single visualization from using the display values and instead use the original values in the column, you can specify a custom expression on that axis. For example:  
`Integer([Employee ID]) as [Employee ID]`

### Example

In this example, you have two different data tables in the analysis; the first one contains the salaries for a number of employees, identified using the employee number:

Salaries		
Employee ID	Salary (thousands of dollars per year)	Data table: Salaries
1354	68	
1355	78	
1356	70	
1357	45	
1358	96	
1359	53	
1360	98	
1361	43	
1362	88	
1363	92	
1364	71	
1365	61	
1366	71	
1367	95	
1368	43	
1369	76	
1370	48	
1371	64	
1372	87	
1373	70	

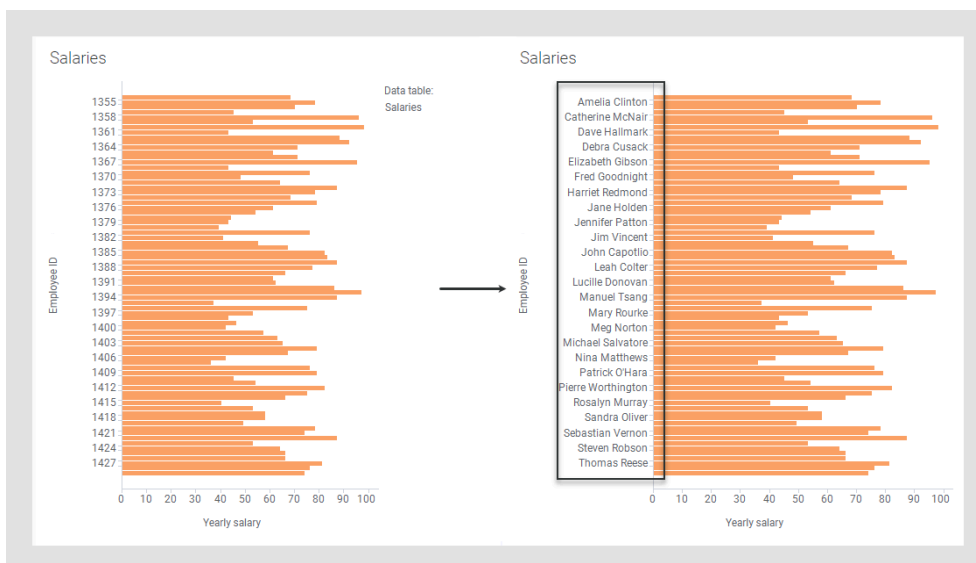
The second data table contains a mapping of the employee number to the first name and last name of the employees:

## Employees

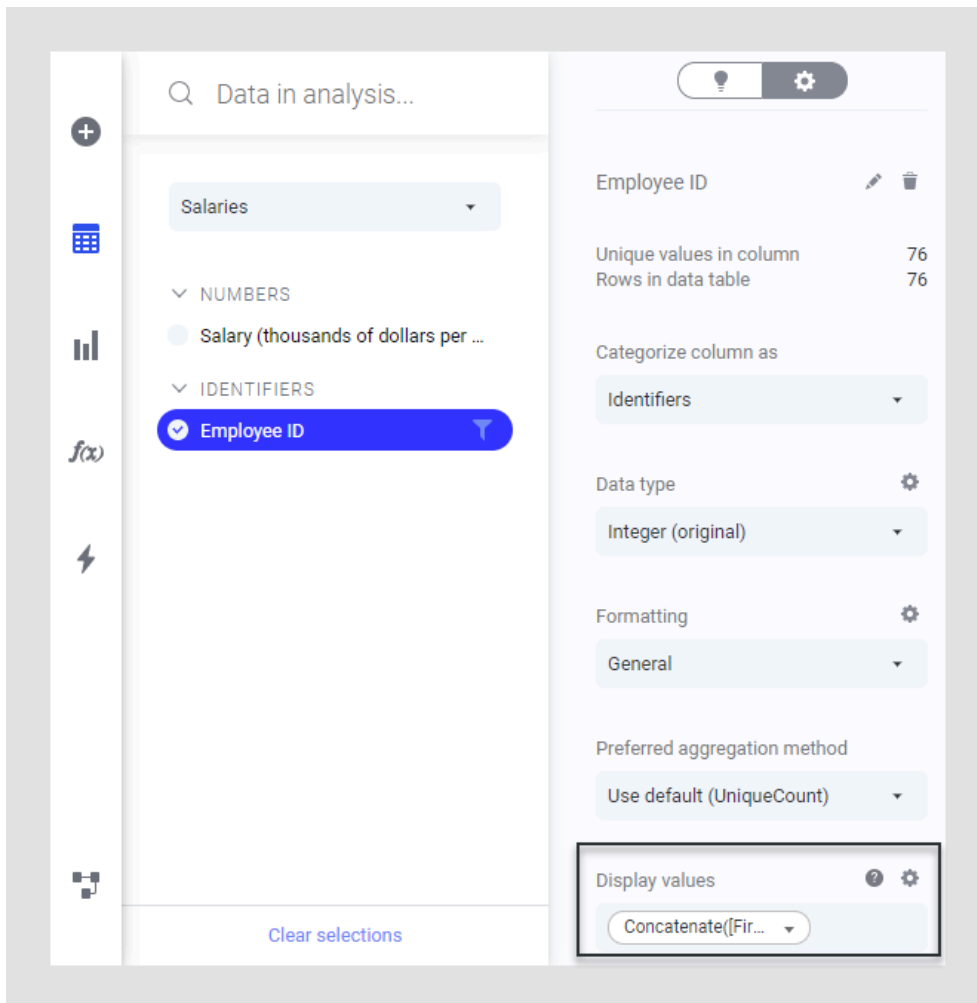
Employee ID	First name	Last name
1354	Adrian	Jones
1355	Amelia	Clinton
1356	Anne	Chang
1357	Bonnie	Foster
1358	Catherine	McNair
1359	Charlotte	Gordon
1360	Daniel	Osterman
1361	Dave	Hallmark
1362	David	Roswell
1363	David	Williams
1364	Debra	Cusack
1365	Diana	Ericsson
1366	Douglas	Moore
1367	Elizabeth	Gibson
1368	Elizabeth	Turnell
1369	Eric	Ballard
1370	Fred	Goodnight
1371	Grace	Milburn
1372	Harold	Kelly
1373	Harriet	Redmond

Data table:  
Employees

When showing the raw data from the salaries data table in a bar chart you would just see a bunch of bars with numbers on them, but by setting the display value for the employee number column to be the concatenated first name and last name of the mapping table (using the custom expression `Concatenate([Employees].[First name], " ", [Employees].[Last name])`) you will see which employee has which salary directly:



The display name for the Employee ID column in the Salaries data table has been specified in the expanded **Data in analysis** flyout:



See also [Sorting columns with display values](#) on page 164.

## Sorting columns with display values

If you define display values, data in visualizations that use them will, by default, be sorted by the alphabetical order of the display values. However, it is also possible to use display values but to show the data sorted by the original values. This gives you the opportunity to, for example, sort the bars in a bar chart by another column, or using a custom expression.






When you sort data based on the display values, this is a purely visual change; any calculations are always based on the original values. If you add any expressions that rely on the order of the data, such as OVER expressions, the sorting will always be based on the original values (regardless of what has been selected in the Display values dialog).

### Prerequisites

See [Setting display values based on other columns](#) on page 159 for generic information about display values.



## Procedure

1. On the [authoring bar](#), click **Data in analysis**  and click on the column of interest.  
That is, select the column that you have modified display values for, and want to update the sorting on.
2. In the expanded flyout, make sure that **Details on selected column**  is selected.
3. In the lower part of the expanded flyout, click the settings button next to **Display values** .
4. In the Display values dialog, select whether to **Sort on display values** or to **Sort on original column values**.



## Splitting a column

It is easy to split a column containing text (a string column) into two or more columns, using the expanded Data in analysis flyout in Spotfire. You can choose to split on any separating character, like space, comma, @, etc. You can also trim away unwanted characters from the original string at the same time.

### Prerequisites

You must have some data loaded in the analysis and the analysis must be in **Editing** mode. The column to split must be a text-based (string) column.

### Procedure

1. On the [authoring bar](#), click **Data in analysis**  and click on the column of interest.
2. In the expanded flyout, make sure that **Details on selected column**  is selected.
3. In the expanded flyout, locate the **Actions** section and click **Split column**.
4. In the Split column dialog, verify that the suggested separator is correct.  
The characters used as separator will be removed from the resulting column values. You can use any character such as ".", " " (whitespace), "@", or a combination of different characters as the separator.
5. Verify that the suggested number of columns is correct.  
If you are only interested in the first (or last) values in the string to split, you can reduce the number of new columns here.
6. Optionally, you may want to remove unwanted characters from the beginning or end of the original string, before performing the split.  
For example, if you have string values within a parenthesis or within quotation marks, you can trim away these characters by specifying a number of characters to remove from the beginning or end of the string.
7. Optionally, if you are more interested in the values at the end of the string, you can click **Split from end of string**, to start counting the substrings to extract from the end instead of from the beginning.  
For example, this can be important if you want to extract information about different countries from a list of email addresses, where the number of different parts included in the addresses vary between different rows. See the Splitting email addresses example below.
8. Click **OK**.

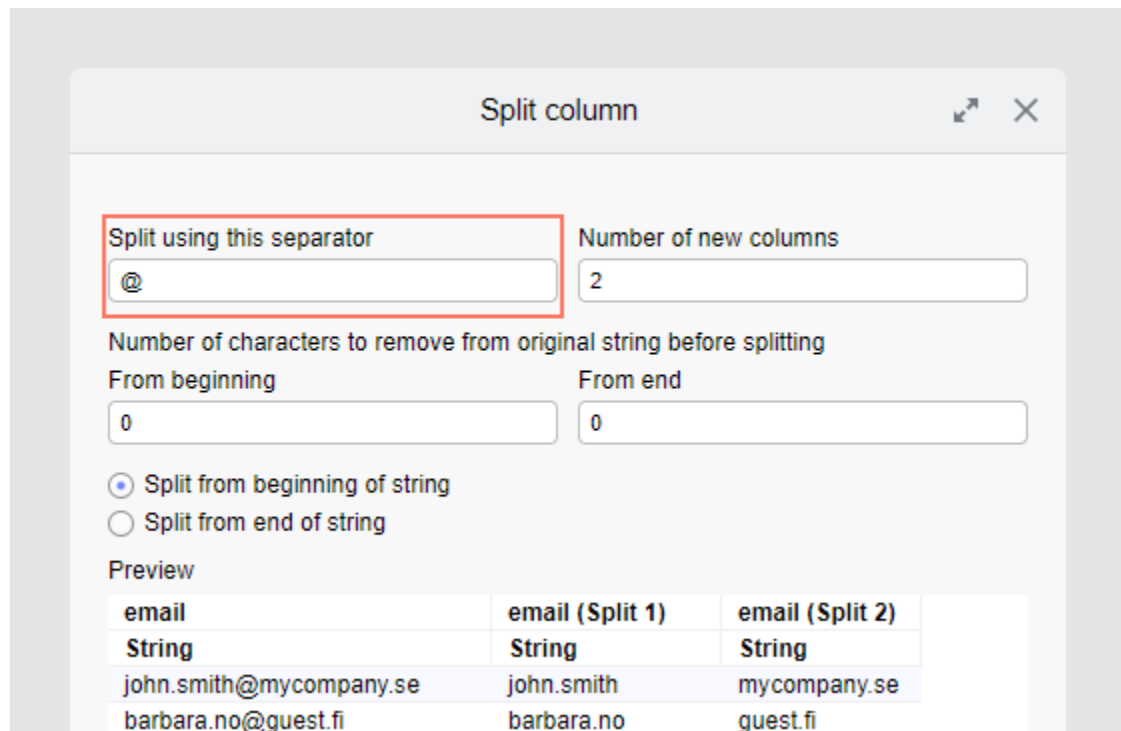
## Result

The new columns show up in the **Data in analysis** flyout. The new columns are added as calculated columns, and it is possible to [edit their name](#) or [expression](#) in the same way as for other calculated columns. For a description of the Split function, see [Text functions](#).

## Splitting email addresses

If you want to split a list of email addresses on the form "firstname.lastname@company.country", it might be difficult for Spotfire to guess what you want to do with your splitting operation, and you may need to do some manual configurations.

If it is the names you are interested in, you might want to start by doing a split using "@" as separator, to get the names in one column, and then do a subsequent split on the new name column using "." as separator, to separate the first name from the last name.



Split column

Split using this separator: @

Number of new columns: 2

Number of characters to remove from original string before splitting

From beginning: 0

From end: 0

☒ Split from beginning of string

☐ Split from end of string

Preview

email	email (Split 1)	email (Split 2)
String	String	String
john.smith@mycompany.se	john.smith	mycompany.se
barbara.no@guest.fi	barbara.no	guest.fi

However, if all you are interested in is the country code, you can instead choose to split on "." directly, set **Number of new columns** to "1", and select **Split from end of string**, to get the country codes only in a new column.

**Split column**

Split using this separator: .

Number of new columns: 1

Number of characters to remove from original string before splitting

From beginning: 0

From end: 0

☐ Split from beginning of string

☒ Split from end of string

Preview

email	email (Split 1)
String	String
john.smith@mycompany.se	se
barbara.no@guest.fi	fi

## Deleting a column

You can remove a column in the expanded **Data in analysis** flyout of Spotfire clients, if you no longer want to include it in the analysis.



When the data comes from an external (in-database) data connection, you will not be able to delete columns from that source. Instead, it may be possible to [edit the connection](#) and exclude the unwanted column from the connection.



Following the steps described below deletes the column from the underlying data table. If you only want to delete a column from a table visualization, but keep the underlying data unchanged, you can [hide the column](#) instead.



### Prerequisites

You must have some data loaded in the analysis and the analysis must be in **Editing** mode.

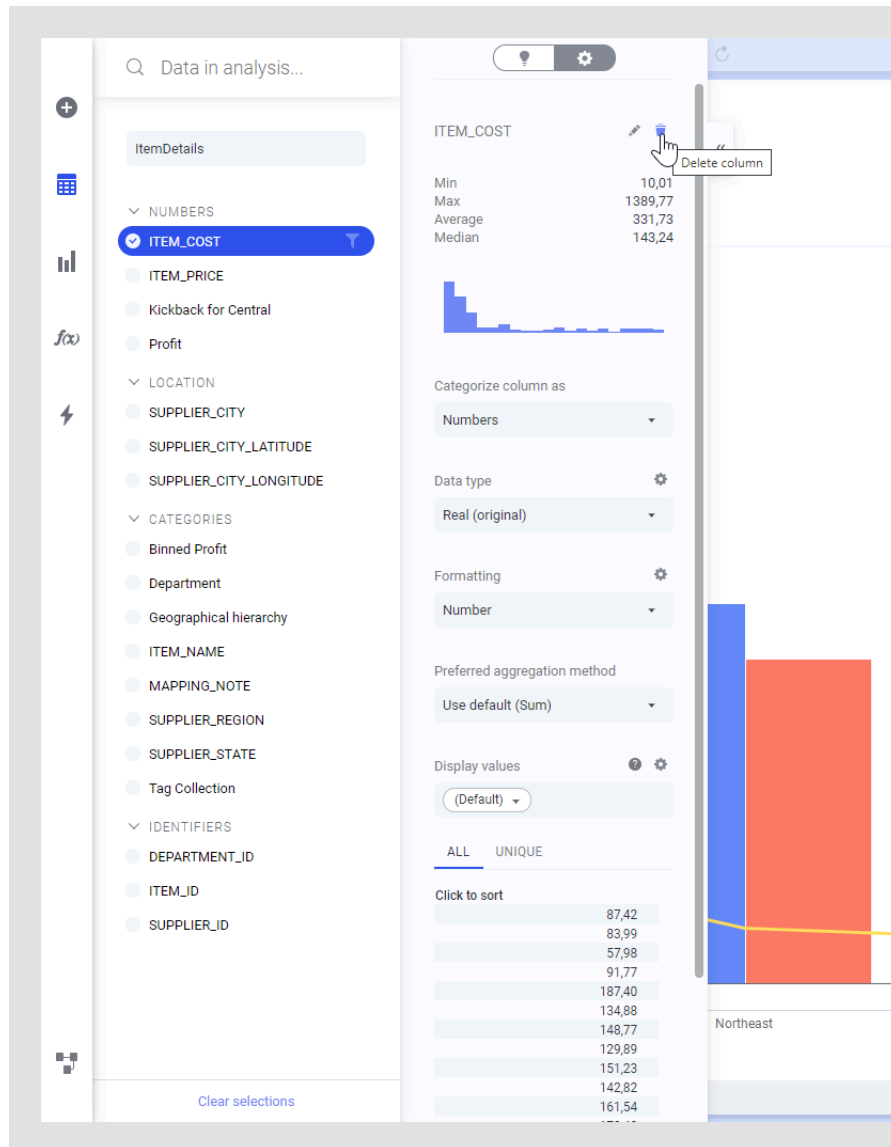


You can also right-click on the column in the **Data in analysis** flyout and select **Delete** from the pop-up menu.

### Procedure

1. On the [authoring bar](#), click **Data in analysis**  and click on the column of interest.
2. In the expanded flyout, make sure that **Details on selected column**  is selected.

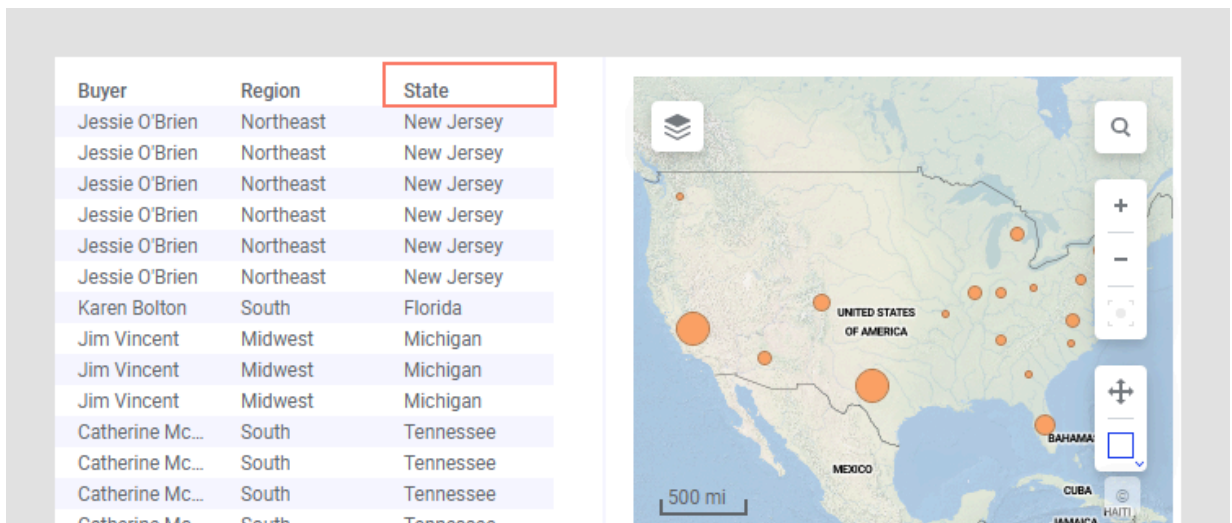
3. At the top of the expanded flyout, click **Delete column** .



## Geographic location and geocoding

To display data on a map, the data must either be geocoded or directly contain coordinates such as longitude or latitude. Geocoding in Spotfire means that some identifiers in a data table are used to match against similar identifiers in another set of data tables (a geocoding hierarchy) which contains latitude/longitude coordinates or geographic features. These coordinates or features are then used for correctly positioning the data in a map context. If your data contains simple geographic elements such as country names, states, or similar, then Spotfire will attempt to automatically geocode your data. If no automatic geocoding can be performed, you can set up the geocoding manually instead.

By specifying that a column in a data table should be used to match against a specific geocoding hierarchy you can make the process of setting up a map chart with that data easier.



In this example, the state column is automatically identified as US States and the data is correctly positioned on the map.

### Automatic geocoding and hierarchies

A geocoding hierarchy is a collection of ordered geocoding tables that can give you the opportunity to drill down to a more detailed level when zooming in on the data. The hierarchy should contain all geocoding tables that the layer should be allowed to choose from when applying geocoding, and it should preferably be a geographic hierarchy with different level of detail, such as:

- Table 1: US States (a table containing state geometries identified by state names (and/or codes).
- Table 2: US Counties (a table containing county geometries identified by the full state/county hierarchy).

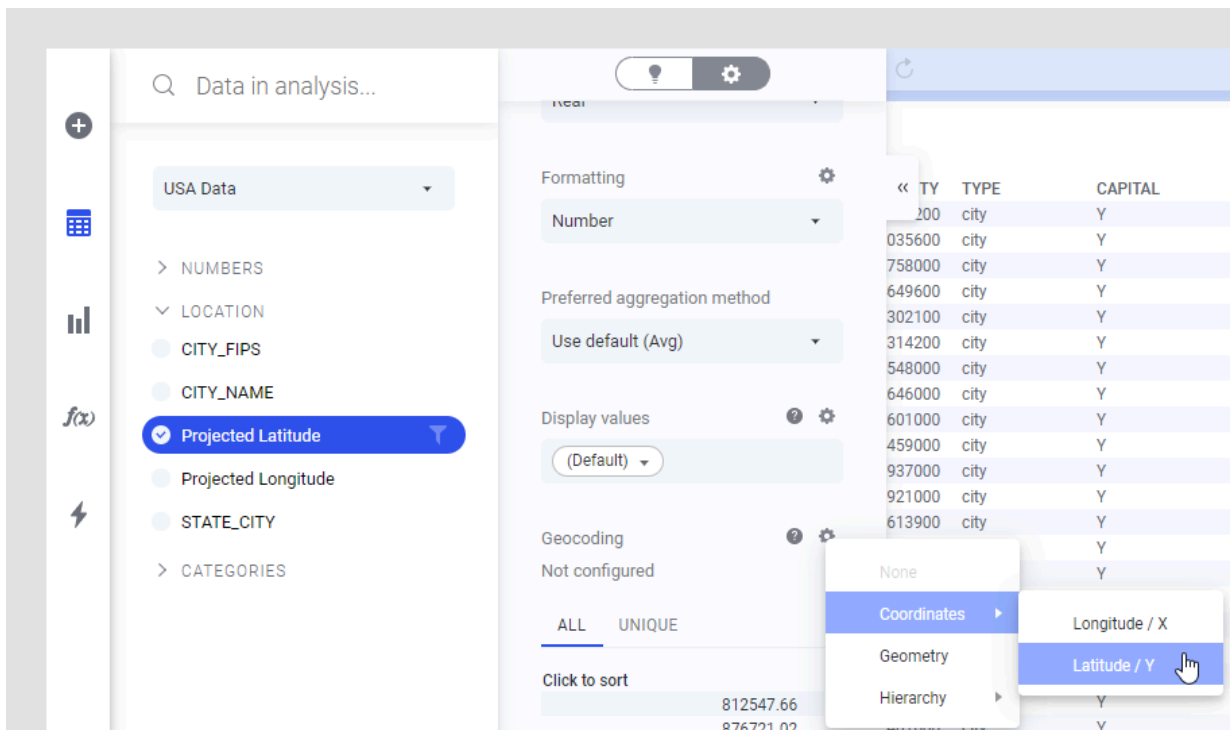
Spotfire automatically adds (hidden) geocoding data tables if you add a map chart for a data table where geographical names can be found. When selecting the best suited geocoding table, Spotfire checks the expression on the Geocode by/Marker by or Feature by-axis and chooses the geocoding table to which the geocoding primary key columns can be matched using column matching. If only State is used on the axis, the US States table would be selected, but if both State and County are used on the axis, then the US Counties table is selected.

Once Spotfire has chosen a geocoding table, the visualization downloads and joins in that table in the map chart based on the found column matches.

### Manual geocoding

When you are manually setting up geocoding you must consider which columns in your data can be used to identify the location of the data. For example, this could be a State column or a State/County hierarchy. Note that the level at which you select to display your data is the one that should be used for geocoding, since the Marker by axis and the Geocoding by axis in map charts are the same in marker layers. Hence, if you want to show data for different states you must geocode by state and not by counties.

You can specify the geocoding type and representation for a selected column in your analysis by clicking on the column in the **Data in analysis** flyout and, in the **Details on selected column** view of the [expanded flyout](#), choosing a suitable **Geocoding** property or hierarchy. You can easily specify that a column contains either longitude or latitude values, geometries, or some hierarchical location. See also [Specifying new geocoding tables](#) on page 172.



You can also try using the **Auto-match** option (available under **Hierarchy** in the **Geocoding** settings) to try to match on the content of the column rather than on the column name, if you know that your data contains common locations but the column names in the data table do not match any of the columns in the available geocoding tables.

### Geocoding hierarchies in the library

A set of default geocoding hierarchies is shipped with Spotfire Server as a zip archive which a library administrator can import into the library. Each hierarchy consists of a number of geocoding tables. Once imported, the geocoding hierarchies are normally located in the GeoAnalytics folder in the library. Because most of these geocoding tables contain geographic shapes, they can also be used as feature layers in your map chart. The geocoding hierarchies located in the library will show up in the list when you choose to add a new marker layer of feature layer in a map chart.

You can also add your own geocoding tables to the library by following the instructions in [Specifying new geocoding tables](#) on page 172.



If you are using Spotfire Analyst without a server, you must install the default geocoding tables using a separate installer, as described in the readme file available with your installation kit. After installation, the geocoding tables are by default located in the folder `C:\Users\<your user>\AppData\Local\Spotfire\<version number>\Example Data\Geocoding Data\`.

### Disabling automatic geocoding in an on-premises environment

A Spotfire on-premises Administrator can disable auto-loading of geocoding tables completely for a group of users by setting the `AutoloadGeocodingTables` preference to `False`. This preference is found in the installed client by selecting **Tools > Administration Manager**. Go to the **Preferences** tab, and look in the `MapChartPreferences` group to locate the preference.

This can be desired if the group is on a slow connection or if the system repeatedly makes unwanted geocoding matchings.

## Specifying new geocoding tables


Spotfire comes with a selection of geocoding hierarchies that are normally stored in the library (for on-premises and TIBCO Cloud Spotfire users). You can also define your own geocoding tables using shapefiles (.shp), GeoJSON files (.geojson), or data tables with categorical (usually `string`) hierarchies, and save them in the library for future use. This is accomplished by setting a few data table and column properties on the geocoding table in Spotfire and then exporting the file to the library. It is also possible to use locally stored geocoding tables.

### Prerequisites

You should have a basic understanding of the concepts in [Geographic location and geocoding](#) on page 169. See also [Configuring WKB data for use with maps](#) on page 174.



You can also specify some geocoding properties for a column from the **Data in analysis** flyout.


Click on the column, make sure that **Details on selected column**  is selected in the expanded flyout and choose suitable options under **Geocoding, Settings**.

In most cases, your analysis will already contain the column properties or data table properties needed, because they are automatically added when opening geographical data. However, if a property that you want to specify is missing, you can add it from the **Data tables** overview in the data canvas. See [Adding a new data table or column property](#) on page 246 for more information.



Different types of geographical data requires a different set of properties and you usually only require some of the properties mentioned below.

### Procedure

1. Load the data containing the coordinates and/or geometries along with the identifying columns (such as state name or state code) in your Spotfire client.
2. On the [authoring bar](#), click **Data canvas** .
3. Make sure the data table of interest is selected, click on the final node of the graphical structure and, in the lower right part of the data canvas, click **Column properties**.

Here, you can specify the necessary column properties to identify columns as geocoding columns.

4. Locate each column that should be used to geocode data in the drop-down list (you can type to search for a column name) and double-click on a property name to open a dialog where you can specify the property value (note that there might be several different columns in your data table that need one or more of these properties specified. For example, if you have a Geometry column, an X column and a Y column, all of those need different values on the `Mapchart.ColumnTypeID` property, as indicated below):

- **GeocodingType** is used to specify the type of geocoding identifiers in the column (such as Country, State or City).
- **GeocodingRepresentation** is used to specify the representation of the geocoding identifiers. It can be a language code (such as `en-us`, `US-English`) or a standard identifier, such as `ISO3166-2:us`, or some other ISO standard code.
- **GeocodingHierarchyName** can be used to group geocoding columns that belong in the same hierarchy.

If the data originates from an ESRI shapefile or a GeoJSON file, the coordinates and geometry columns will automatically be tagged correctly, otherwise you must add the following column properties, as specified below (or see [Configuring WKB data for use with maps](#) on page 174):

- A longitude column should have the property **MapChart.ColumnTypeID** with the value `XCenter`.



- A latitude column should have property **MapChart.ColumnTypeID** with the value `YCenter`.
- A geometry column should have property **MapChart.ColumnTypeID** with the value `Geometry`.
- A geometry column should also have the property **ContentType** with the value `application/x-wkb`.

These property values are what will be matched with the same property values of the data that should be geocoded later, so choose wisely.

- When you have specified the required column properties, click **Data table properties** to specify one or more of the following data table properties:
  - **MapChart.IsGeocodingTable** is used to determine whether a saved SBDF file should be interpreted as a geocoding table or not. The data type for this property must be set to `Boolean`. Set the value for this property to `True`. The export will then pick up that it is a geocoding table and automatically transfer the properties to library properties.
  - **MapChart.GeocodingHierarchyName** is used to group geocoding tables that belong in the same hierarchy. The data type is normally a `String`. Values are, for example, "USA 2022", "Belgium", and so on.
  - **MapChart.GeocodingAutoload** is used to indicate whether the geocoding table should be allowed to be automatically downloaded to the analysis when Spotfire finds out that it can be used as a geocoding table for the current data table. This value can be used to prevent large geocoding tables from being automatically downloaded. The data type must be `Boolean`. The value must be either `True` or `False`.
  - **MapChart.IsGeocodingEnabled** is used to indicate whether the geocoding table is currently enabled. The data type must be `Boolean`. The value must be either `True` or `False`. This property is important when geocoding tables are saved in the library.
  - **MapChart.GeometryType** is used to define the type of feature when the geocoding table contains a Geometry (shape) column. The data type must be `String`. The values can be either `Point`, `Line` or `Polygon`.

There are also a number of additional properties that you can specify for geocoding tables. These properties are not mandatory:

- **MapChart.GeocodingHierarchyPriority** is used to control the priority of the hierarchy in which the table is included. A geocoding hierarchy with higher priority will be selected first if two hierarchies result in an equal number of column matches when automatic geocoding is applied. The data type must be `Integer`, and the default value must be between 0 and 100 (100 being the highest priority).
  - **MapChart.GeocodingHierarchyVersion** is used to specify the version of the geocoding hierarchy in the form of a date (YYYYMMDD).
  - **MapChart.GeographicCrS** is used to specify the coordinate reference system describing the data (for example, "EPSG:4326" for WGS 84).
  - **MapChart.ExternalGeographicCrS** is used to store the EPSG code of the projection identified by the shapefile projection web service. The `MapChart.ExternalGeographicCrS` property is in use when the identified projection already exists in the Spotfire list of projections.
  - **MapChart.ExternalGeographicProj4Def** is used to store the Proj4 definition generated by the shapefile projection web service. The value stored in this property is used by Spotfire to create a Generated CRS. The `MapChart.ExternalGeographicProj4Def` property is in use when the shapefile's projection does not already exist in the Spotfire list of projections.
- Export the table to the library as SBDF using **File > Export > Data to library** to make it available for other users and for use in other analyses.  
The data table is saved as a file in the library and will be available in all places where a geocoding hierarchy can be selected.

## Configuring WKB data for use with maps

When geographical data is located in a Well-Known Binary (WKB) column, but not originating from a shapefile or GeoJSON file, you can manually create the required coordinate columns using the spatial functions in the expression language in the add calculated column tool.

See also [Spatial functions](#) on page 889, [Geographic location and geocoding](#) on page 169 and [Specifying new geocoding tables](#) on page 172.

### Procedure

1. Select **Data > Add calculated column**.
2. Add the first spatial function, `WKBEnvelopeXCenter`, to the expression.
3. Select the WKB column from your data table as the argument for the function, for example, `WKBEnvelopeXCenter([WKB column])`.
4. Type a suitable **Column name** (e.g., `XCenter`) and click **OK**.  
The `XCenter` column is created and the correct property value is set on the **MapChart.ColumnTypeId** property.
5. Repeat steps 2 to 4 with the other five Spatial functions.
6. Now, create a map chart and use the data table with the newly created columns as the data table in the **Data** section of the Feature Layer Settings.

## Unpivoting columns


Data can be organized in different ways, for example, in a short/wide or tall/skinny format, but still contain the same information. Often, it is easier to visualize data organized in a tall/skinny format, that is, when the values are collected in just a few value columns. Unpivoting is one way to transform data from a short/wide to a tall/skinny format, so the data can be presented the way you want it in the visualizations.

You can select several columns with similar values and combine them into a single column which includes all values, using **Unpivot**. If the data types of source columns should differ, the varying data is converted to a common data type in the combined column.

### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.

### Procedure

1. On the authoring bar, click **Data in analysis**  and click to select the columns to combine.
  - To select a consecutive group of columns, click the first column, press Shift, and then click the last column.
  - To select non-consecutive columns, press Ctrl, and then click each separate column, or, make sure to select the check box for each column.
2. Right-click on one of the chosen columns and select **Unpivot** from the pop-up menu.

### Result

The values from the selected columns are combined into a single column called "Value". Also, a new category column is created, where the previous column names are used as categories.

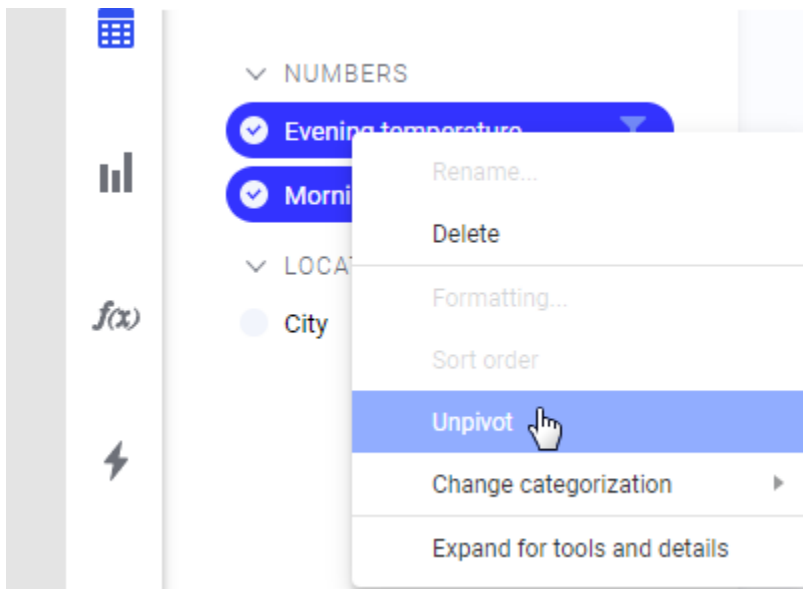
### Example: Combining multiple temperature inputs to an average temperature

This example shows unpivoting of a very simple data table. In the original data table, there are three columns and four rows. Each row contains a city, a morning temperature and an evening temperature for each city:

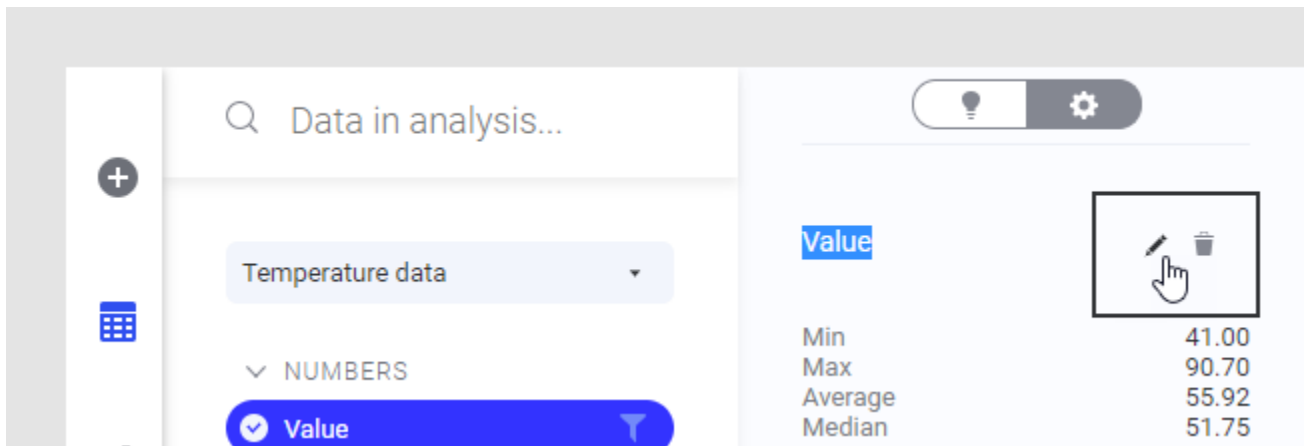
City	Morning temperature	Evening temperature
Austin	62	90.70
Boston	41	48.00
Chicago	51	57.20
Denver	45	52.50

While this is certainly useful data, you may want to determine the average temperature of all the cities for the whole day instead.

In the **Data in analysis** flyout, click to select the Morning temperature and Evening temperature columns, then right-click and select **Unpivot**.



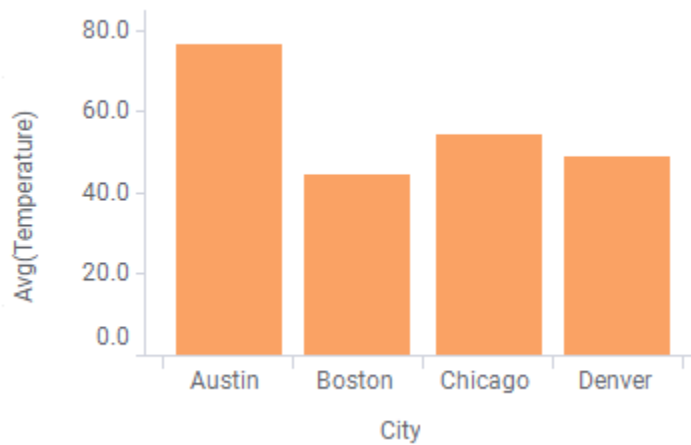
You can [change the names](#) of the new columns in the flyout, if desired. In this example, the name of the new value column is changed to "Temperature", and the new category column is called "Type of measurement".



After unpivoting the data, there is one row for each measurement in the data table.

City	Type of measurement	Temperature
Austin	Morning temperature	62.0
Austin	Evening temperature	90.7
Boston	Morning temperature	41.0
Boston	Evening temperature	48.0
Chicago	Morning temperature	51.0
Chicago	Evening temperature	57.2
Denver	Morning temperature	45.0
Denver	Evening temperature	52.5

Now, you can easily show an average value of the temperature in each city, for example, in a bar chart.



In the original data, the morning temperatures were given as integers and the evening temperatures as real numbers. In the unpivoted data table, all values must be of the same data type. Therefore, the integers were automatically changed to real numbers (because changing the real number temperatures into integers would have resulted in a loss of information).

### Example: Combining sales data from several desks into a single column

In this example, a larger data set containing data on the sales of entrance tickets for a museum is used. The original data table shows data for each of the five ticket counters (desks) and the number of tickets they have sold to adults, children and senior citizens each day. The data is organized in a short/wide format, that is, the spreadsheet has many columns with similar data.

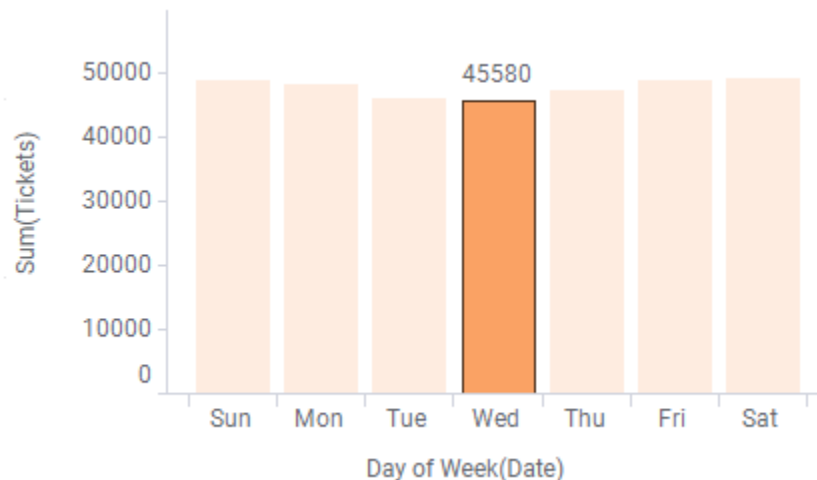
Category	Date	Desk1	Desk2	Desk3	Desk4	Desk5
Adult	5/1/2017	62	19	26	111	34
Child	5/1/2017	20	6	5	11	8
Senior	5/1/2017	102	47	42	6	49
Adult	5/2/2017	74	35	37	4	65
Child	5/2/2017	17	8	8	0	10
Senior	5/2/2017	122	39	47	3	50
Adult	5/3/2017	91	43	24	1	24
Child	5/3/2017	16	8	5	6	1
Senior	5/3/2017	99	35	34	40	55
Adult	5/4/2017	51	21	13	49	104
Child	5/4/2017	16	4	7	5	15
Senior	5/4/2017	64	29	29	51	80
Adult	5/5/2017	135	55	51	37	21
Child	5/5/2017	15	6	4	6	21
Senior	5/5/2017	71	26	23	85	64
Adult	5/6/2017	95	39	38	3	131
Child	5/6/2017	23	10	9	13	1
Senior	5/6/2017	58	18	22	92	87
Adult	5/7/2017	90	40	24	20	134
Child	5/7/2017	23	7	6	19	3
Senior	5/7/2017	117	38	32	48	15
Adult	5/8/2017	104	43	38	46	51
Child	5/8/2017	18	7	5	6	14
Senior	5/8/2017	113	32	44	48	45
Adult	5/9/2017	103	26	46	1	51
Child	5/9/2017	13	4	4	10	14
Senior	5/9/2017	89	43	41	53	60
Adult	5/10/2017	55	23	26	90	22
Child	5/10/2017	20	8	8	0	7
Senior	5/10/2017	91	26	32	4	111

If you are more interested in analyzing ticket sales in general, rather than needing to know which counter sold how many tickets to whom, you can unpivot the data. This way, you can combine the Desk columns into a single column, and merge all ticket sales numbers to another column.

In the data table below, the same data is organized in the tall/skinny format. The values from the desks have been combined in one column, called "Tickets", and "Desk" is the new category column showing from which desk the tickets were sold.


Category	Date	Desk	Tickets
Adult	5/1/2017	Desk3	26
Adult	5/1/2017	Desk4	111
Adult	5/1/2017	Desk5	34
Adult	5/1/2017	Desk2	19
Adult	5/1/2017	Desk1	62
Child	5/1/2017	Desk3	5
Child	5/1/2017	Desk4	11
Child	5/1/2017	Desk5	8
Child	5/1/2017	Desk2	6
Child	5/1/2017	Desk1	20
Senior	5/1/2017	Desk3	42
Senior	5/1/2017	Desk4	6
Senior	5/1/2017	Desk5	49
Senior	5/1/2017	Desk2	47
Senior	5/1/2017	Desk1	102
Adult	5/2/2017	Desk3	37
Adult	5/2/2017	Desk4	4
Adult	5/2/2017	Desk5	65

By doing an analysis of the new data table, you can show that Wednesdays are the days when we sell the least amount of tickets.




## Data canvas

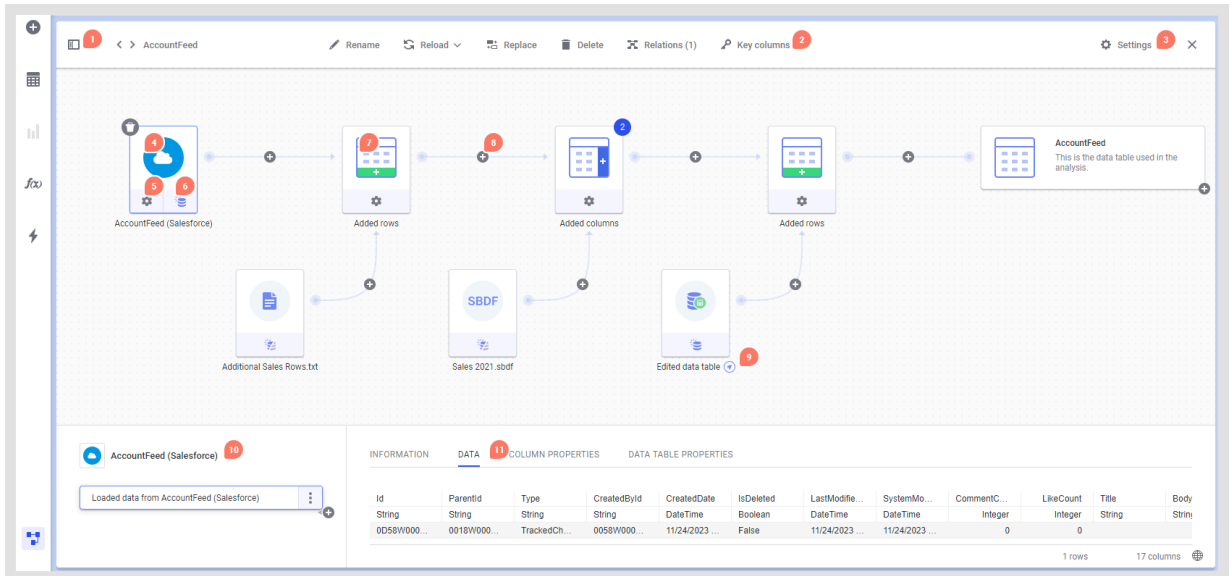
The **Data canvas** (the source view) shows how data has been added to the analysis, and whether any modifications have been done to obtain the final data tables. Each object in the upper part of the data canvas is called a node, and it represents a data source, an operation where data from two sources have been combined, a data function, or similar. In the source structure, you can click on different nodes to see what has been done with the data in that step. This can help you get an insight in how the final data table, used in the analysis, was created. You can also use the data canvas to change settings for different nodes or operations, or to reload data from different sources.

On the [authoring bar](#), click **Data canvas** .



The toolbar at the top of the data canvas always applies to the final data table, and not to the selected node in the graphical structure. This means that a replace operation from the toolbar will replace everything done with the data table, up until the final node. Click on a data source node and select **Replace data source** from the 'Loaded data' step in the lower part of the source view to switch an individual data source. Similarly, 'Delete' on the toolbar will remove the entire data table. To remove a separate node (data source), click **Remove**  on the node of interest instead.

In this example, a data table has been loaded from Salesforce and more rows and columns have been added to the original data table from data files and a linked copy of another data table.

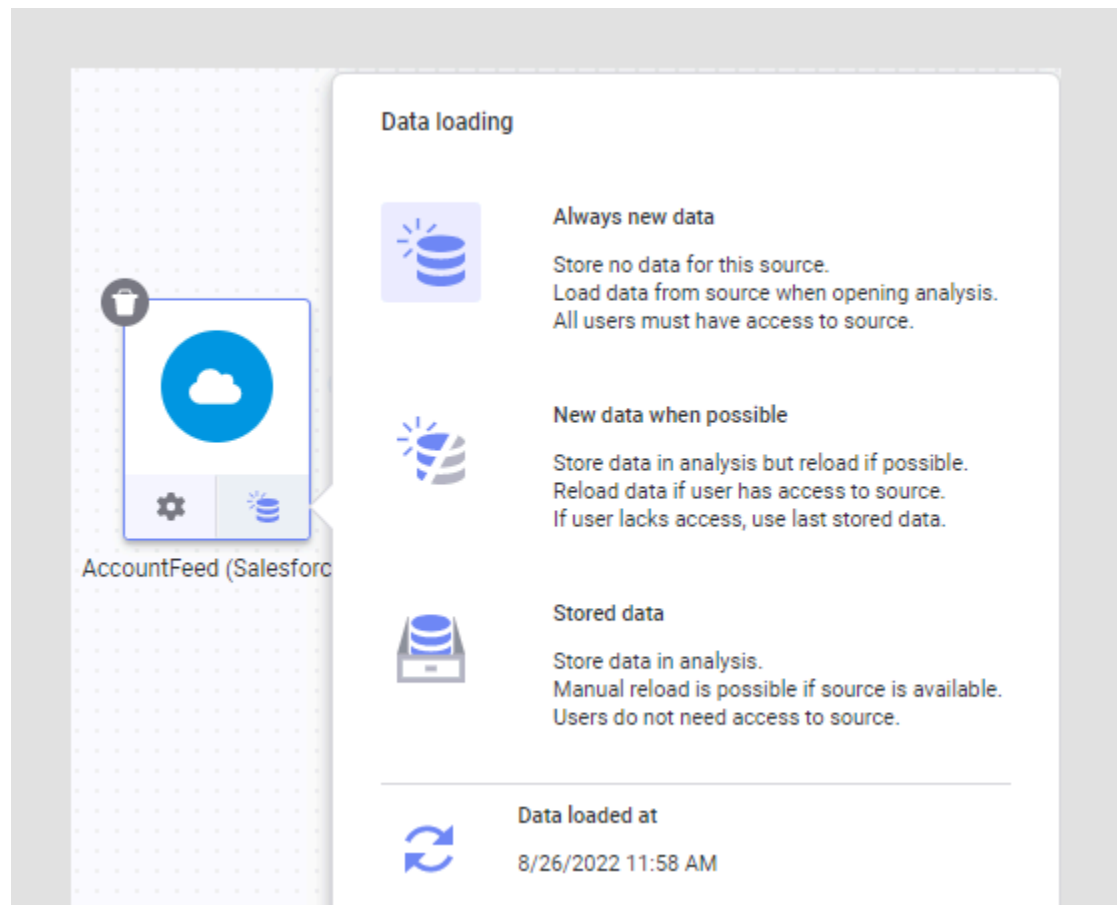


The screenshot shows the Spotfire Web Client interface. The top toolbar includes buttons for Rename, Reload, Replace, Delete, Relations (1), and Key columns (2). The data canvas displays a workflow starting with 'AccountFeed (Salesforce)', followed by 'Added rows' (receiving input from 'Additional Sales Rows.txt'), 'Added columns' (receiving input from 'Sales 2021.sdbf'), and another 'Added rows' (receiving input from 'Edited data table'). The final node is 'AccountFeed', which is highlighted as the current data table. Below the canvas, the 'AccountFeed (Salesforce)' data table is displayed with 1 row and 17 columns. The table structure is as follows:

Id	ParentId	Type	CreatedById	CreatedDate	IsDeleted	LastModifie...	SystemMo...	CommentC...	LikeCount	Title	Body
0D58W000...	0018W000...	TrackedCh...	0058W000...	11/24/2023 ...	False	11/24/2023 ...	11/24/2023 ...	0	0		

1. Switch which data table or data function to inspect using the data canvas sidebar, or use the arrows to step through the list of all data tables or all data functions, depending on what is selected. You can use search in the sidebar if your analysis contains many different items. The top **Data tables** header in the sidebar can be selected to show an overview of all data tables in the analysis, and their properties. See [Data tables overview](#) on page 201 for more information.
2. The toolbar at the top of the data canvas contains tools that work on the entire data table, such as **Rename**, **Reload**, **Replace**, **Delete**, **Relations** and **Key columns**.
3. If you click **Settings** on the data table toolbar, you get access to more settings that affect the whole data table. For example, here you reach the option to [top-level embed](#) the final data table, change how to handle **Filters** for the data table, and you have the option to hide data tables (for example, if they are used for intermediate calculations) from axis selectors by clearing the **Show data table in user interface** check box. You can also choose to **Cache calculated columns** or (in the installed client) clear the **Show prompt for column mismatches** check box.
4. In the graphical view of the data canvas, click on a node to view more information about that node in the fields below.
5. If the settings for the node can be changed, click the settings icon under the node to edit it. See [Editing a data connection](#), [Editing column matching or other settings for added rows](#) and [Editing settings for added columns](#) on page 189 for more information. Note that settings buttons that you have had access to in a previous session might be missing if data from another node in the data table is unavailable, so make sure that you get all of the expected data into the analysis.

6. For some types of sources, you can change the **Data loading** settings by clicking on the button at the bottom of a node, and switching to a different [setting](#):



Here, you can also reload linked data, for data sources where this is possible, as described in [Reloading data](#).




Some nodes do not have access to the original source. In that case, the only option available will be Stored data. For example, this is the case when you have uploaded a file from your local computer to the Spotfire web client, or if you have added data from the clipboard using the installed client. See [Linked, stored, and embedded data](#) on page 70 for more information about data loading settings.

7. When more data has been added to the original data table, the data to add is shown as a separate node and the original data plus the added rows or columns is shown in the **Added rows** or **Added columns** node.
8. You can add more rows or columns to a certain point in the data structure by clicking the plus sign between two nodes. When using the installed client, you can also add transformations between nodes. Transformations added using the plus sign will end up as the last step on the node to the left of the plus sign. To add a transformation between existing step groups (installed client only), see 10.
9. Navigation arrows can be shown if a data table is a [linked copy of another data table](#), or if the data comes from a data function. Click the arrow to navigate to the linked item in the data canvas.
10. The field at the bottom left shows the list of step groups on the currently selected node, and, when applicable, provides an entry point for [replacing a data source](#) or for [removing previously added operations](#). Using the installed client, you can also [add](#) or [edit transformations](#) or [edit on-demand settings](#) here. If you want to add more rows or columns to a certain point in the structure, you can click the plus sign of interest and select the desired operation. Note that if you insert rows or columns between transformation groups in the list of step groups, any transformations below the



insertion point will be moved to the new node in the data structure. If there is an issue with an operation, you will be informed about this using icons and tooltips.

11. The field to the right shows details about the selected node or operation. For in-memory sources, click the **Data** header to see a preview of the data the way it looks after the steps in the selected step group have been applied. You can also see the number of rows and columns in the data at this stage, which makes it possible to investigate whether data has been added in the expected way. The **Column properties** and **Data table properties** headers shows the value for each column or data table property, respectively, as specified for the selected node or operation. You can edit column or data table property values on the final node of the data table, as specified in [Editing data table or column property values](#) on page 247.
12. Switch between the data canvas and the visualization canvas by clicking the **Data canvas** icon  on the authoring bar.

The node to the far right represents the final data table, the way it is used in visualizations in Spotfire. The plus on the final data table allows you to add things like [calculated columns](#) or [hierarchies](#), and in the installed client also [binned columns](#).

If data functions have been added, you can view information about their current configuration and edit their inputs and outputs from the Data canvas. Click the navigation arrow to jump between the data table source view or the data function view in the data canvas. For more information about data functions, see [Data functions in the Data canvas](#) on page 668.

## Changing the name of a data table

The name of a data table is shown in all data table selectors and it is used as the default name in new table visualizations. If the original data table name is unclear, you might want to change it to a better name. Authors can rename data tables in the **Data canvas**.


### Prerequisites

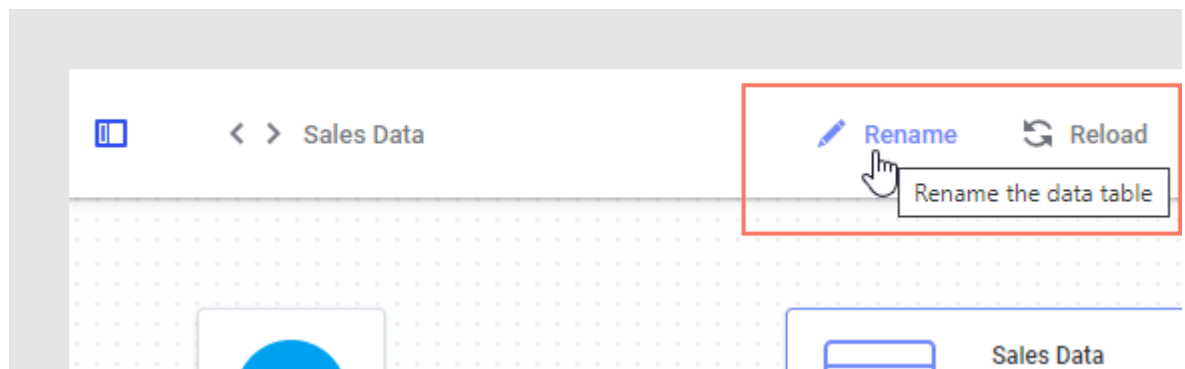
You must have some data loaded in the analysis and the analysis must be in **Editing** mode.



When using the installed client, you can also rename the data table by right-clicking on the data table selector in the **Data in analysis** flyout and selecting **Rename**.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.
3. On the data canvas toolbar, click **Rename**.



4. In the popover, type a new name for the data table and click **OK**. You cannot have two data tables with identical names in the same analysis.  
The data table name is updated.

## Deleting a data table

If you have added more than one data table to your analysis, you can delete unused data tables from the data canvas.



When there is only one data table left in the analysis, you cannot delete it. However, you can always [replace the data table](#) instead.





You can delete multiple data tables at once from the **Data tables** overview in the [Data canvas](#) on page 178 sidebar.

### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.
3. On the data canvas toolbar, click **Delete** .




Any visualizations that used a deleted data table must be updated manually.

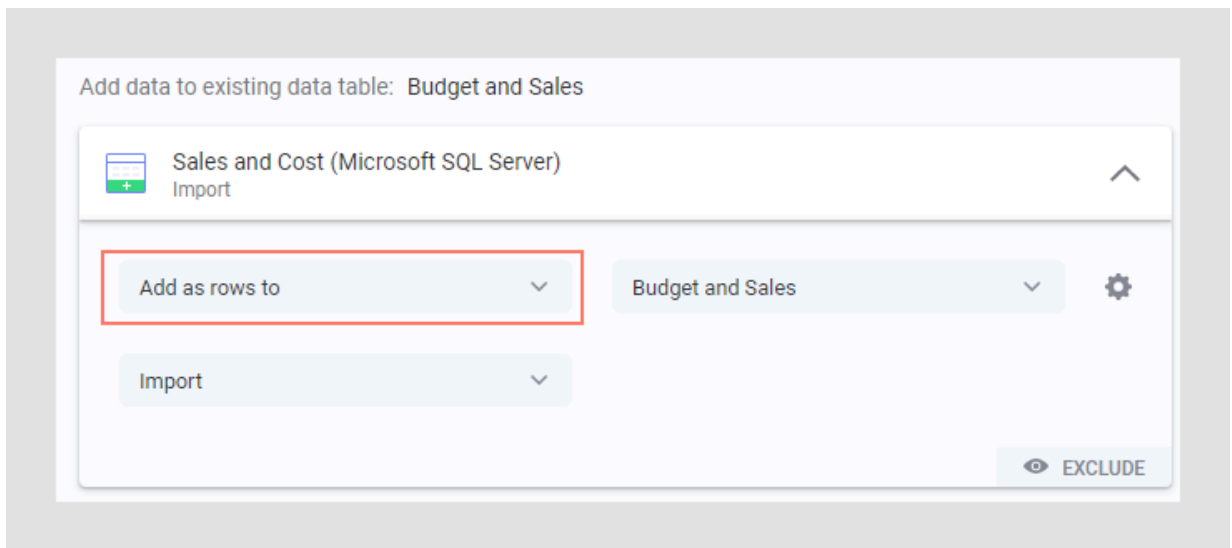
## Adding rows to a data table

If you are working with imported (in-memory) data, you can add rows from a different source to your current data table. Data can be added from a local file, a data file previously saved in the Spotfire library (using the installed client), from data functions that produce data, from data connections saved in the library, or you can add data directly from Google Analytics or from Salesforce, if you have access to those systems. In the installed client you can also add data directly from other data sources using connectors, from information links, or from the clipboard.

Even though data from different data tables often can be viewed in the same visualization, it might be preferred to actually incorporate similar data into a single data table. For example, if all columns are the same in the data sources and you know that you want to use summarized values from all sources in your visualization.

You add data from the **Files and data** flyout  or from the **Data canvas**.

When you add data from the **Files and data** flyout, you can select to add the data as new rows in another data table from the final step in the add data workflow:




You can also insert rows into a specific place in a data table structure, as described below.

### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.
3. In the data table structure, click on the plus sign between the nodes where you want to insert rows, and select **Add rows**.



You can also add rows between previously added transformation groups for a selected node.

4. Select data to add rows from.
5. In the flyout, click **Settings for added rows**.
6. In the Add rows – match columns dialog, verify that the suggested column matching is valid, or edit the matching as desired. This is done by adding or removing matches between columns **From original data** and **From new data**.

Click on the plus sign and select a column from the list to add a match for one of the columns from the original data. Click on the x on a row to remove a match.

Alternatively, you can select **Use auto-match** to automatically match on name and data type, if you know that these are the same in both sources. However, with auto-match, Spotfire will automatically match columns on name and data type each time the data is reloaded and manual matching of columns will be unavailable.



When adding rows, you will append new rows to the original data table. This means that you should probably try to match as many columns as possible. See the preview for a hint on how data will be matched.

7. Optionally, if there are additional columns in the added data, you can select to include or skip these columns under **Include additional columns from new data** by selecting or clearing the check box for each column.



If the data source from which data was added can be reloaded, and new columns become available after a reload, then the new columns will automatically become included. If this is not desired, you may want to exclude new columns after reloading data by [editing the settings for added rows](#).

8. Optionally, change the settings under **Identify origin of rows in a column**.  
Select **Do not identify origin** to skip this option, **Create new column** to add a new column with different values for data from the original source and data from the added source, or select **Use existing column** to add this information to an existing column.
9. When you are done, click **OK**.

### Example: Adding rows from Salesforce

In this example, data has already been opened from a local Excel file. In the **Data canvas**, select the data table of interest, determine where you want to add rows (between the nodes in the graphical structure or between transformation groups), and click **Add rows**:

The screenshot shows the Data canvas interface. At the top, there's a data table titled 'My Company Sales - North America' with a description 'This is the data table used in the analysis.' Below it, a list of transformations is shown: '1. Calculate new column' and '2. Exclude columns'. A red box highlights the 'Add rows' button in the 'Add rows | Add column' flyout menu. The 'DATA' tab is selected, showing a table with columns 'Number' and 'Sales'.

Number	Sales
Integer	Real
2004	2691.88
2005	910.51
2006	69.15

7085 rows

Choose a new or a predefined connection to Salesforce and sign in. [Select the data](#) to add and click **OK**. In the Add rows – match columns dialog, verify that the suggested column matching is valid, or add or remove matches between columns **From original data** and **From new data**. When you are satisfied with the data selection and settings, click **OK** to close the flyout and add the data.

### Editing column matching or other settings for added rows

If you have added rows from a different source to a data table, there might be occasions when you must update the settings used for matching the data from different sources together. For example, a column used for matching linked data can have been renamed or removed in either of the sources.




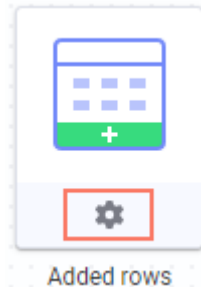
In the [Data canvas](#) on page 178, you will see a red exclamation mark on the **Added rows** node, if the current settings need updates.

## Prerequisites

Rows must have been added to a data table in the analysis. See [Adding rows to a data table](#) for information on how to add rows. The analysis must be in **Editing** mode.

## Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.
3. In the **Data canvas**, click the settings button on the **Added rows** node.



4. In the Settings for added rows dialog, verify that the column matching is still valid, or edit the matching as desired. This is done by adding or removing matches.  
Click on the plus sign and select a column from the list to add a match for one of the columns from the original data. Click on the x on a row to remove a match. If an earlier match is broken, click on the exclamation mark and select a new match, when possible, or remove the match. Alternatively, you can select **Use auto-match** to automatically match on name and data type, if you know that these are the same in both sources.



When adding rows, you will append new rows to the original data table. This means that you should probably try to match as many columns as possible. See the preview for a hint on how data will be matched.

5. Optionally, if there are additional columns in the added data, you can select to include or exclude these columns under **Include additional columns from new data** by selecting or clearing the check box for each column.



If the data source from which columns were added can be reloaded, and new columns become available after a reload, then the new columns will automatically become included. If this was not desired, you may want to exclude new columns after reloading data.

6. Optionally, change the settings under **Identify origin of rows**.  
Select **Do not identify origin** to skip this option, **Create new column** to add a new column with different values for data from the original source and data from the added source, or select **Use existing column** to add the information about where rows come from to an existing column.

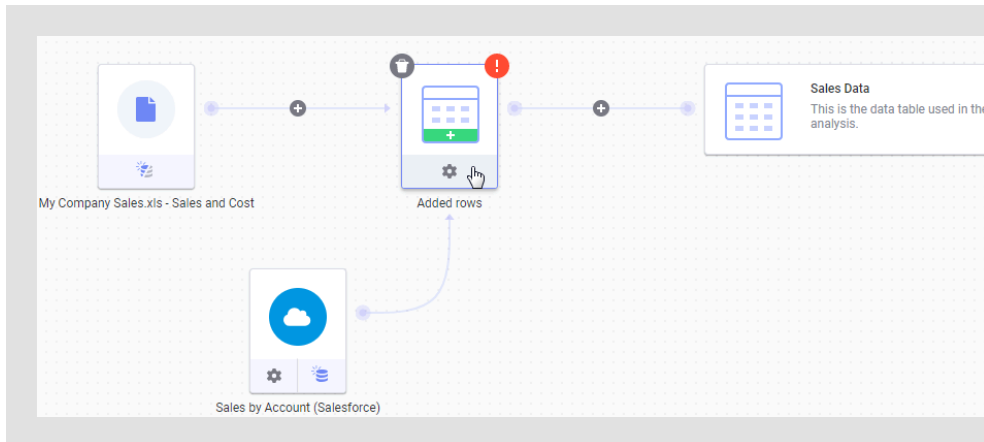


If a previously used column has been removed or renamed, you might need to select a different option before you can choose a different, existing column here.

7. When you are done, click **OK**.

### Example: Editing added rows when matching is broken

In this example, rows have been added from Salesforce to a data table that was originally opened from a local Excel file. The local file was saved using the data loading setting "New data when possible" and it has been reloaded at some time. Before the Excel file was reloaded it was modified; the name of the column used in the matching was changed. Therefore, the previous matching of columns is now broken, which is indicated with a red exclamation mark in the **Data canvas**:



To fix the broken match, first click the settings icon on the **Added rows** node. This opens the Settings for added rows dialog, and the broken match is indicated with red exclamation marks:

Match columns
2 of 7 columns matched

Type to search in list

From original data	From new data	
Delivery Date (Date)	Close Date (String)	!
Number (Integer) 2004, 2005, 2006		+
Sales (Real) 2691.88, 910.51, 69.15	Amount (String)	x
Cost (Real) 1442.00, 639.12, 71.47		+
Category (String) Spices, Fruit, Vegetables		+
Type (String) Saffron, Plums, Apples	Type (String)	x
Order Date (Date) 12/28/2013, 12/26/2013, 12/27/2...		+
Close Date (Date) 12/31/2013, 12/30/2013, 12/29/2...		+

Click on the red exclamation mark and then click **Remove match**. Add a new match for the renamed column in the original data by clicking on the plus sign next to the new column name and selecting a matching column from the new data.

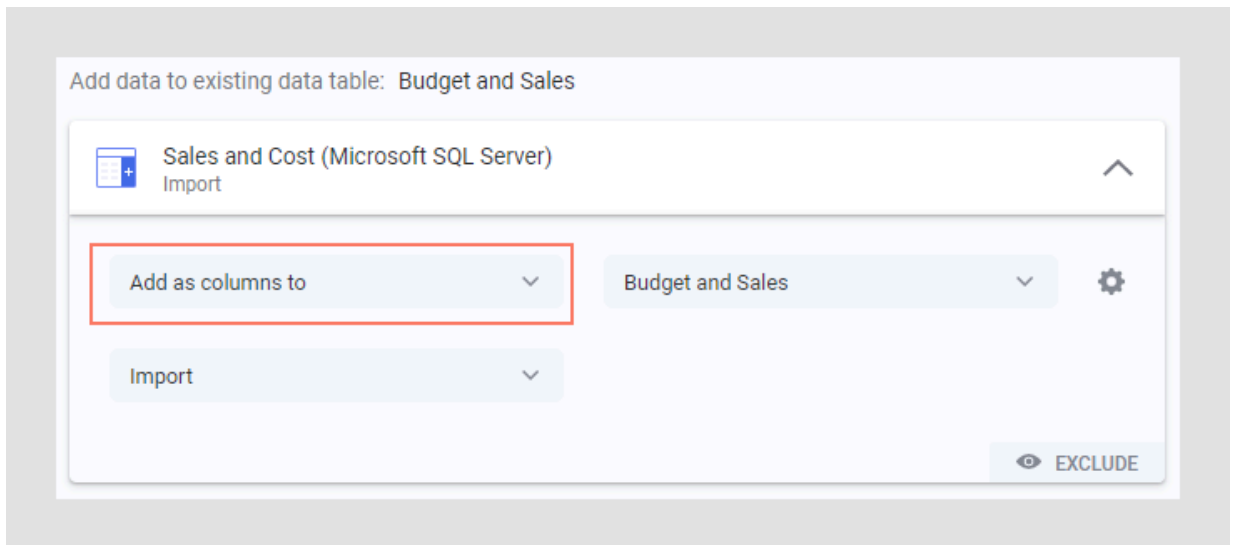
## Adding columns to a data table

If you are working with imported (in-memory) data, you can add columns to your current data table to create a join between data from different sources and merge it to a single data table. Data can be added from a local file, a data file previously saved in the Spotfire library (using the installed client), from data functions that produce data, from data connections saved in the library, or you can add data directly from Google Analytics or from Salesforce, if you have access to those systems. In the installed client you can also add data directly from other data sources using connectors, from information links, or from the clipboard.

Even though data from different data tables often can be viewed in the same visualization, it might be preferred to actually incorporate similar data into a single data table. When multiple data tables are used in one visualization, the join of the two tables is done after aggregation has been performed. Therefore, if you want to perform joining of tables before any aggregations, or, if you want to create certain calculated columns based on data from two sources, you might want to insert data from one table to the other.

You add data from the **Files and data** flyout  or from the **Data canvas**.

When you add data from the **Files and data** flyout, you can select to add the data as new columns in another data table from the final step in the add data workflow:




You can also insert columns into a specific place in a data table structure, as described below.

### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.

3. In the data table structure, click the plus sign between the nodes where you want to add columns, and select **Add columns**.



You can also add columns between previously added transformation groups for a selected node.

4. Select data to add columns from.
5. In the flyout, click **Settings for added columns**.
6. In the Add columns – match columns dialog, verify that the column match suggested by the recommendation engine is the way you want it, or configure the matching as desired. You can add or remove matches, or edit an existing match. To edit a match, click on the row with the match and select one column **From original data** and one **From new data**. Click on the x on a row to remove a match.



When adding columns, matching is done to enable joining between two data sets. Typically, only one or a few columns containing row identifiers should be used in the match.



If you cannot match your desired columns directly, a transformation added on the node of the original data using the installed client might be used to create a calculated column to use in the match.

7. To the right in the dialog, you see a preview of the data, using the current settings. You can change the **Number of input rows** to include more rows (from both input tables) in the sample calculation. This can sometimes increase the accuracy of the shown result. However, using too many rows can decrease the performance of the preview.
8. Optionally, if you do not wish to add all available columns from the added data, as is the default, you can select to skip columns under **Columns from new data** by clearing the check box for the undesired columns.



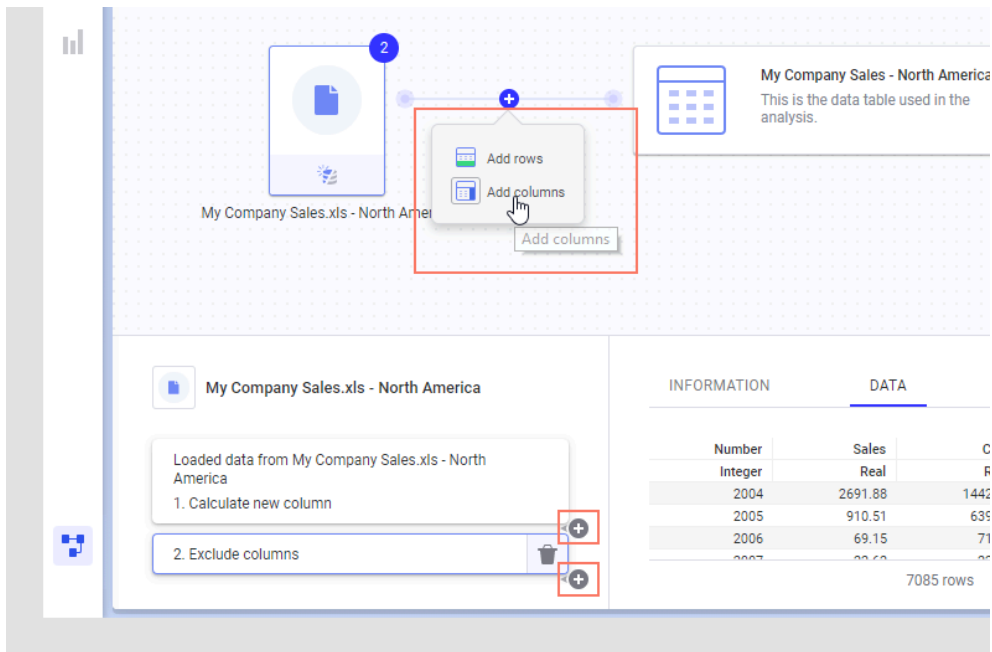
If the data source from which columns were added can be reloaded, and new columns become available after a reload, then the new columns will automatically become included. If this is not desired, you might want to exclude new columns after reloading data by [editing the settings for added columns](#).

9. Review the **Join settings**. Look at the interactive Join example, at the bottom of the Join settings field, for information about how rows will be matched using the different join types. You can specify whether or not to match on empty values using the **Treat empty values as equal** check box. Note that selecting an inner join can result in that no data remains in Spotfire, if no matching rows are found. You can edit the join settings at a later stage from the data canvas.
10. When you are done, click **OK**.



### Example: Adding columns from Salesforce

In this example, data has already been opened from a local Excel file. In the **Data canvas**, select the data table of interest, determine where you want to add columns (between the nodes in the graphical structure or between transformation groups), and click **Add columns**:



Choose a new or a predefined connection to Salesforce and sign in. [Select the data](#) to add and click **OK**. In the Add columns – match columns dialog, verify that the suggested column matching is valid, or add or remove matches between columns **From original data** and **From new data**. When you are satisfied with the data selection and settings, click **OK** to close the flyout and add the data.

See also [Adding a calculated column](#) on page 730.

### Editing settings for added columns

If you have added columns from a different source to a data table, there might be occasions when you need to update the settings used for matching the data from different sources together. For example, a column used for matching linked data can have been renamed or removed in either of the sources, or the selected join type might not provide the expected result.




In the [Data canvas](#) on page 178, you will see a red exclamation mark on the **Added columns** node, if the current settings need updates.

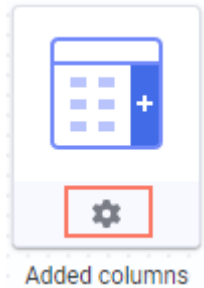
#### Prerequisites

Columns must have been added to a data table in the analysis. See [Adding columns to a data table](#) on page 187 for information on how to add columns. The analysis must be in **Editing** mode.

#### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.

3. In the **Data canvas**, click the settings button on the **Added columns** node.



The Settings for added columns dialog shows a preview of the data, using the current settings. You can change the **Number of input rows** to include more rows (from both input tables) in the sample calculation. This can sometimes increase the accuracy of the shown result. However, using too many rows can decrease the performance of the preview.

4. In the Settings for added columns dialog, verify that the column matching is still valid, or edit the matching as desired. This is done by editing, adding or removing matches.

If an earlier match is broken, an exclamation mark is shown on the row. To edit a match, click on the row with the match and select one column **From original data** and one **From new data**. Click on the x on a row to remove a match. You can also add a new match.



When adding columns, matching is done to enable joining between two data sets. Typically, only one or a few columns containing row identifiers should be used in the match.

5. Review the columns available under **Columns from new data** and modify which columns to include or exclude by selecting or clearing the check box for each column.



If the data source from which columns were added can be reloaded, and new columns become available after a reload, then the new columns will automatically become included. If this is not desired, you might want to exclude new columns after reloading data.

6. Review the **Join settings**.

Look at the interactive Join example, at the bottom of the Join settings field, for information about how rows will be matched using the different join types.

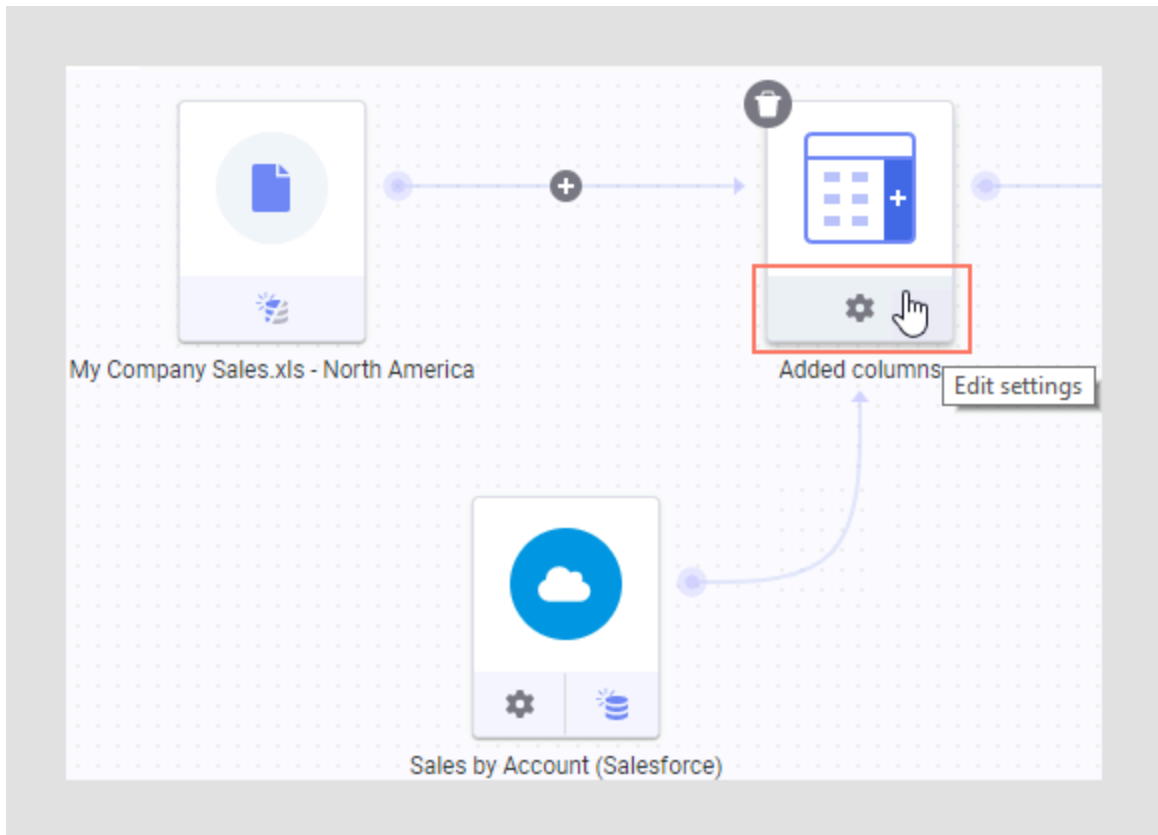
You can specify whether or not to match on empty values using the **Treat empty values as equal** check box.

7. When you are done, click **OK**.

### Example: Editing the Join settings for added columns

In this example, columns have been added from Salesforce to a data table originally based on a local Excel file. After analyzing the combined data table, it was discovered that not all of the expected rows from the Salesforce data were available. In this case, it is possible to fix the issues by editing the join settings for the added columns.

In the **Data canvas**, click the settings icon on the **Added columns** node:



This opens the Settings for added columns dialog. By inspecting the **Join settings** section and trying out the interactive join example, it was concluded that the currently selected join type, left outer join, was unsuitable for the current use case. With a left outer join, rows that are available in the second data table only will not be included in the resulting data table:



☒ **Right outer join**  
*Rows from new data only*

☐ **Right single match join**  
*Rows from new data only, single match from original*

☐ **Treat empty values as equal**

---

Join example (Right outer join; no match on empty values)

A	B
a	1
b	2
c	3
c	4
	5

A	C
a	6
c	8
c	9
d	10
	11

A	B	C
a	1	6
c	3	8
c	4	8
c	3	9
c	4	9
d		10
		11

Hover with the mouse pointer over rows to see how the selected join type works.


## Storing data within the analysis

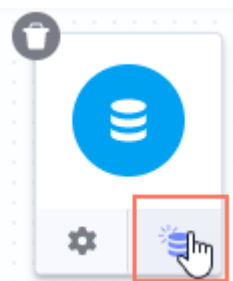
When you are working with data from a linkable data source (e.g., data from Salesforce or saved data files in the library), you can decide whether to save the data within the analysis or whether the data should be reloaded from the source each time you open the analysis.

### Prerequisites

You must have some data loaded in the analysis and the analysis must be in **Editing** mode.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.
3. In the data canvas, locate the node that represents the data source of interest and click the **Data loading** button at the lower part of the node.



4. In the **Data loading** pop-over, select the **Stored data** option.  
The **Stored data** option saves the current data in the analysis. New data will only be loaded if the source is manually reloaded. The **New data when possible** option will also store the current data in the analysis. In that case, data will also be available to users who do not have access to the source.


However, if a user has access, then new data will be loaded when the analysis is opened. **Always new data** does not store any data in the analysis file.

## Removing operations in the data canvas

Some operations on your data can be removed in the data canvas.

For example, if a transformation has been added using the installed client but it is no longer applicable, you can remove it. Or, if you have deleted rows or columns from the analysis and you want to bring the missing data back, you can remove the delete-operation. To remove data sources or added rows/columns, see [Removing a data source from a data table](#) on page 195.

### Procedure


1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.
3. In the data canvas, click on the node where the transformation or operation you want to remove was applied.

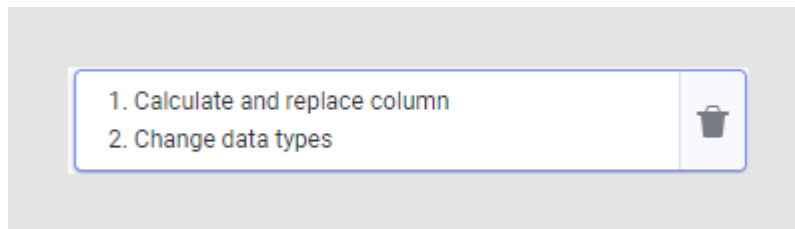
If there is an error in a transformation you will be guided to the affected node and transformation group by error icons.




If the transformation was added on a data loading-step, you will not be able to remove it from the web client. However, in an installed client, you can choose to edit the transformation and remove one or more transformations from the data loading-step.

Details about the selected node are shown in the bottom left part of the data canvas.

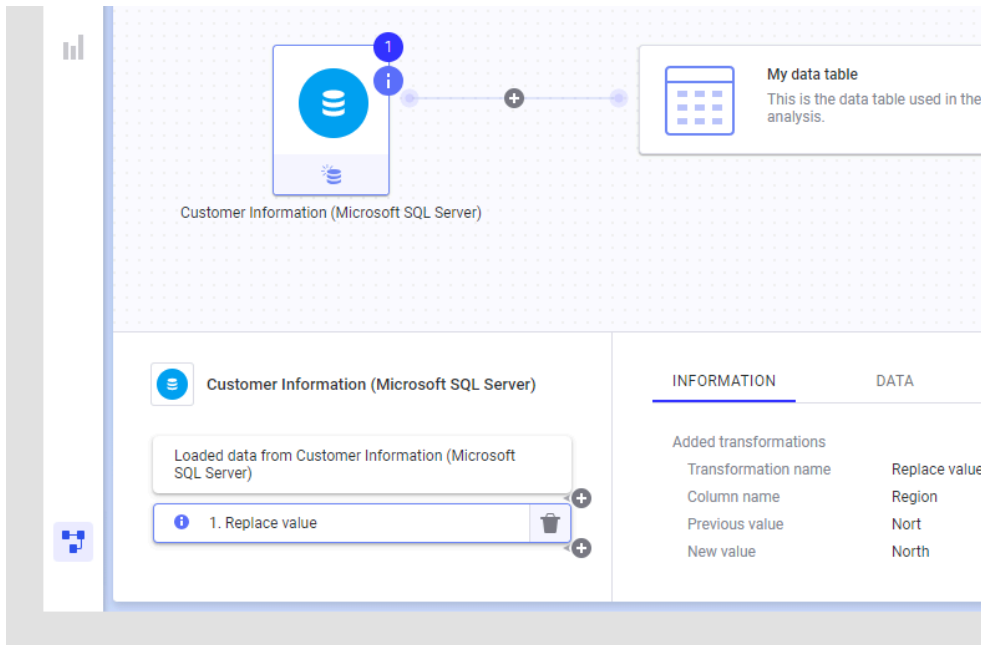
4. For the operation you want to remove, click **Remove** .



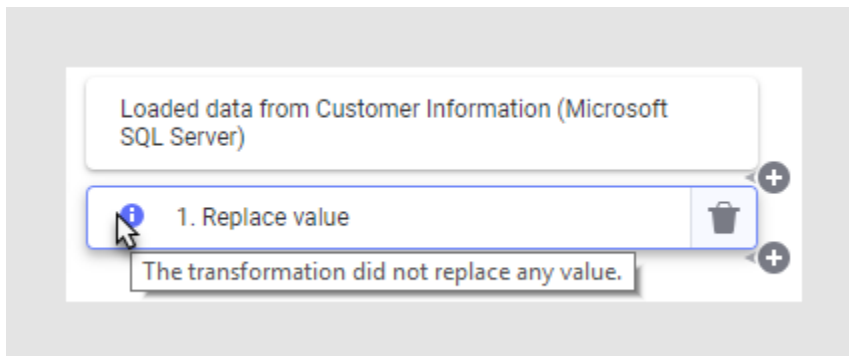
In the installed client, if you have added multiple transformations in one step (within the same transformation group), and you only want to remove one of them, click **Edit transformations**  instead. In the Transform data dialog you can both remove transformations and insert new transformations to the group.


### Example: Removing a 'replace value'-operation that is no longer applicable

In this example, the creator of the analysis used the 'replace value'-transformation to fix a spelling error in the data table. However, at a later stage, someone updated the data in the linked source and the 'replace value'-operation was no longer needed:



There is an information icon on the node, as well as on the 'replace value'-operation in the step list. A tooltip gives you information about the issue:



Click  to remove the operation.

### Removing a data source from a data table


If your data table was built on data from more than one data source (by addition of columns or rows), you can remove data sources from the data table as they become obsolete. This is done in the data canvas.

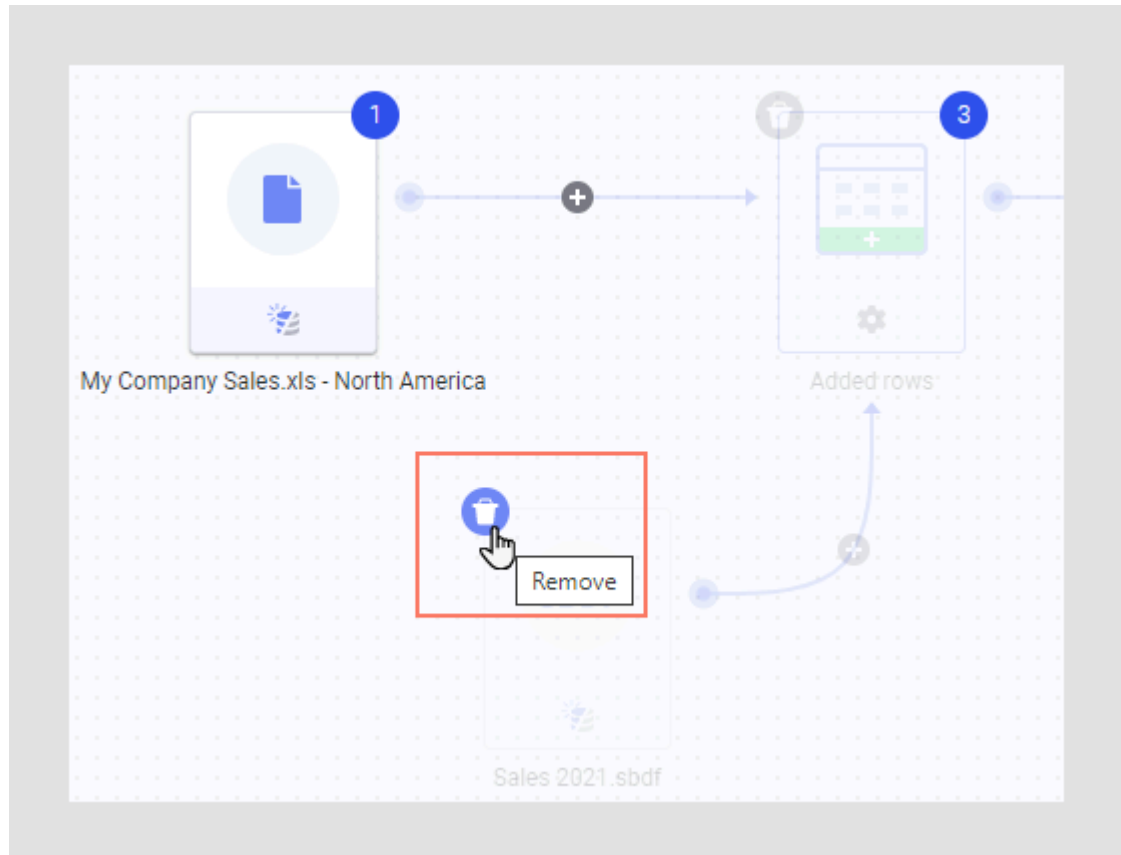
You can remove most data sources (not sources in top-level embedded data tables, sources added before frozen nodes, or when remaining nodes contain no data). However, you cannot remove all data sources in the data table. Instead, you can [remove the entire data table](#).

## Prerequisites

You must have some data loaded in the analysis and the analysis must be in **Editing** mode.


## Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.
3. In the data canvas, hover with the mouse pointer over the data source node or the add rows/columns node that you want to remove.



An icon with a trash can is shown at the upper left corner of the node.

Hovering with the mouse pointer over the icon before clicking it will show you an animation of what will be removed from the data table; for example, a data source and the add rows operation it belonged to.

4. Click **Remove** .
- A confirmation message is shown.
5. Click **OK**.

## Result

The data source and any related add columns or add rows operations are removed from the data table. Note that you might end up with a different set of columns than you had before when you change the structure of the data.

If a data source with transformations on the source node itself is removed, then the transformations will be discarded. Transformations on an **Added rows** node or on an **Added columns** node will



be transferred to the nearest remaining node. If desired, you can always remove any remaining transformations at a later stage, see [Removing operations in the data canvas](#) on page 194.

## Editing a data connection

If you have added data to the analysis using a connection to an external data source (e.g., Salesforce or Google Analytics) you might encounter situations when you want to change which columns to fetch, modify the date range to look at, or, in certain cases, add or remove views (data tables) in the data connection. Then, you can use the **Data canvas** to edit the connection.

### Prerequisites


You must have some data loaded in the analysis and the analysis must be in **Editing** mode.

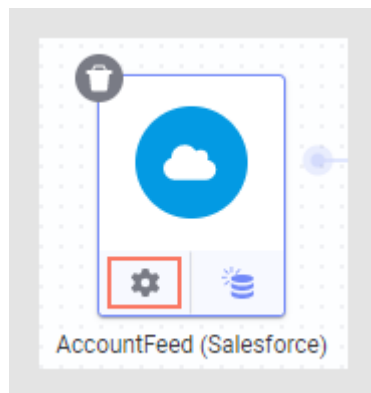
The connection you want to edit must be embedded in the analysis. To edit data connections and data sources that are saved in the library, use [Manage Data Connections](#) from the installed client.



More connectors are available for editing when using the installed client. There, you can also perform changes to the data source settings in the connection, such as changing the authentication method, editing timeout settings, or moving from a test database to a production database.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.
3. In the data canvas, locate the node that represents the data source of interest and click the **Edit settings** button at the lower part of the node.



4. Make the desired changes to the data and click **OK** when you are done.



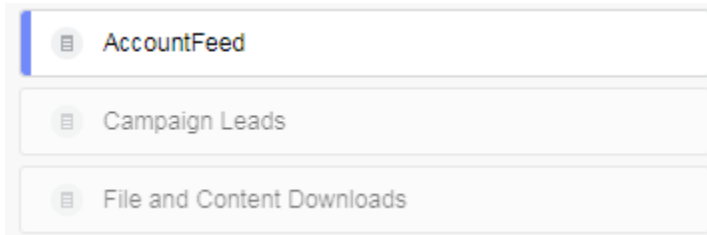
Some data connections (e.g., data from Salesforce) allow you to edit more than the view you are currently working with if you click **Enable full editing of connection**. By doing this, your changes may affect other data tables than the one you are currently looking at. You can also add or remove views in the connection. When using the installed client and full editing has been enabled for connections to relational databases, you can edit the data source settings by clicking **Edit data source settings**.

### Result

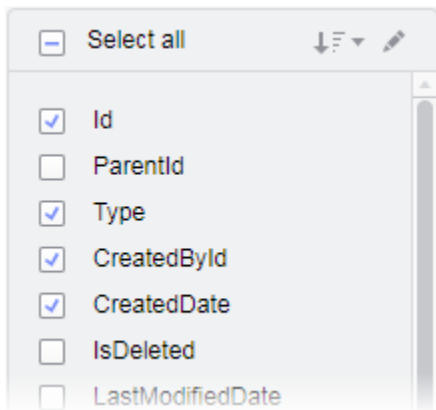
The analysis is updated with the edited data.

### Example: Editing a Salesforce connection

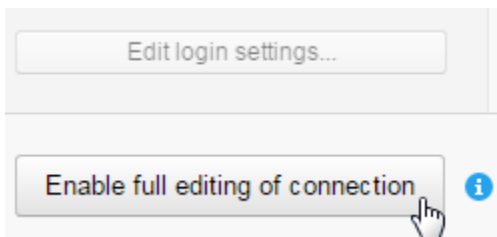
In this example, source data from Salesforce is edited. When clicking the **Edit settings** button in the data canvas, the Edit data from Salesforce dialog is shown (or Views in Connection in the installed client). It is basically the same dialog as the one used when [Accessing data from Salesforce](#) on page 104. When opened from this context, the only view available for editing is the one used to add the data used in the selected node:



You can click to add columns that had been previously left out, or clear the check boxes for columns that you no longer want to include in the analysis.



If you want to make bigger changes to the entire Salesforce connection, click **Enable editing of full connection**.



When the full connection is enabled for editing, you can make changes to other views than the one you started out with. You can also add more views to the connection (which will be added as new data tables in the analysis), or, you can remove views.

While full editing enables **Edit Data Source Settings** in the installed client, the web client instead enables the **Edit login settings** button, which gives you the possibility to change the username and password or to use a sandbox URL instead of the regular Salesforce instance.

## Specifying key columns for a data table

Key columns (or a primary key) are used when you need to uniquely identify all rows in the data table. You should specify key columns if you want to be able to reapply markings in unaggregated visualizations that were active when saving the file, if you want any specified tags or bookmarks to be able to be reapplied when reopening the analysis file, or, if you want to be able to replace specific values in a data table. However, there is no guarantee that a selection always can be reapplied even if key columns are specified because selection of a visualization item might also include references to other columns than the key columns.




Each row must be determined by a unique combination of values in the specified columns.

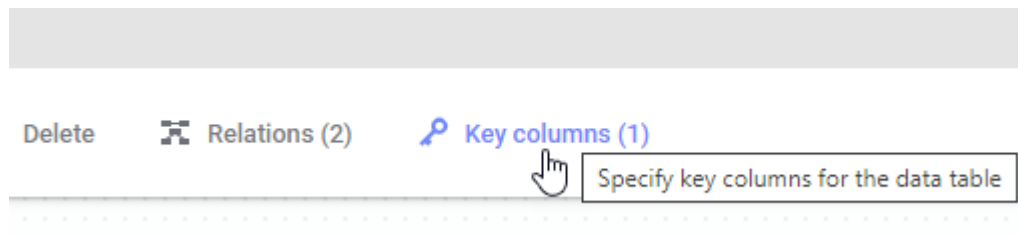
As an example, for each row with a tag or a marking in your current analysis, the values for the specified columns are noted in the saved analysis file, and when the analysis file is opened again, rows matching those criteria will be tagged or marked again. This means that if a new row has been added to the data table that also matches a criterion for a tag or a marking, the tag or marking is not unique and therefore invalid. Neither the new row nor the original row that was tagged or marked, will receive any tag or marking when reopening the analysis. However, a 'Replace specific value'-operation will replace the value for all matching rows, even if key columns no longer are unique. Unless the transformation has been configured to ignore warnings, you will be informed about the non-unique keys in the data canvas.

### Prerequisites

You must have imported data in the analysis. The use cases for this functionality are not applicable for external data.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.
3. On the data canvas toolbar, click **Key columns**.



4. In the Key columns dialog, select one or more columns that uniquely identify all rows in the data table by clicking on them and then clicking **Add >**.  
For details about the **Limit available columns to** drop-down list and the **Hide columns that are not valid for Replace Value** check box, see inline help or the example below.
5. If you need to remove one or more columns from the **Selected columns** list, select the columns and click **< Remove** or click **< Remove all**.
6. When you are done, click **OK**.

### Example: Adding key columns for a customer data table

In a data table where transactions from different customers (buyers) are analyzed, the names of customers are sometimes the same, even though they are different people. To uniquely identify each customer, you might need to specify both the name and the address or region of the customer as key columns.

When you open the Key columns dialog, you might only see recommended columns in the **Available columns** list, but you can use the **Limit available columns** to list to help you find columns to use.

There can be three different options available in this list:

- **Recommended columns**

Lists columns with Integer/LongInteger or String data types that have unique values for all rows, if available. It is likely that these columns are good choices for determining a unique identifier for each row, but you might have a data table where other columns are more suitable. For example, if you have a column containing transaction numbers, it might be suitable to identify specific transactions uniquely, but not to find unique buyers.

- **Columns of appropriate data types**

This option only shows columns of data type Integer/LongInteger or String, because these are more likely to provide unique identifiers. Even if a column does not have unique values of all rows, a combination of several columns (e.g., 'First name' and 'Last name', or 'Name' and 'Address') can be sufficient to provide a unique identification.

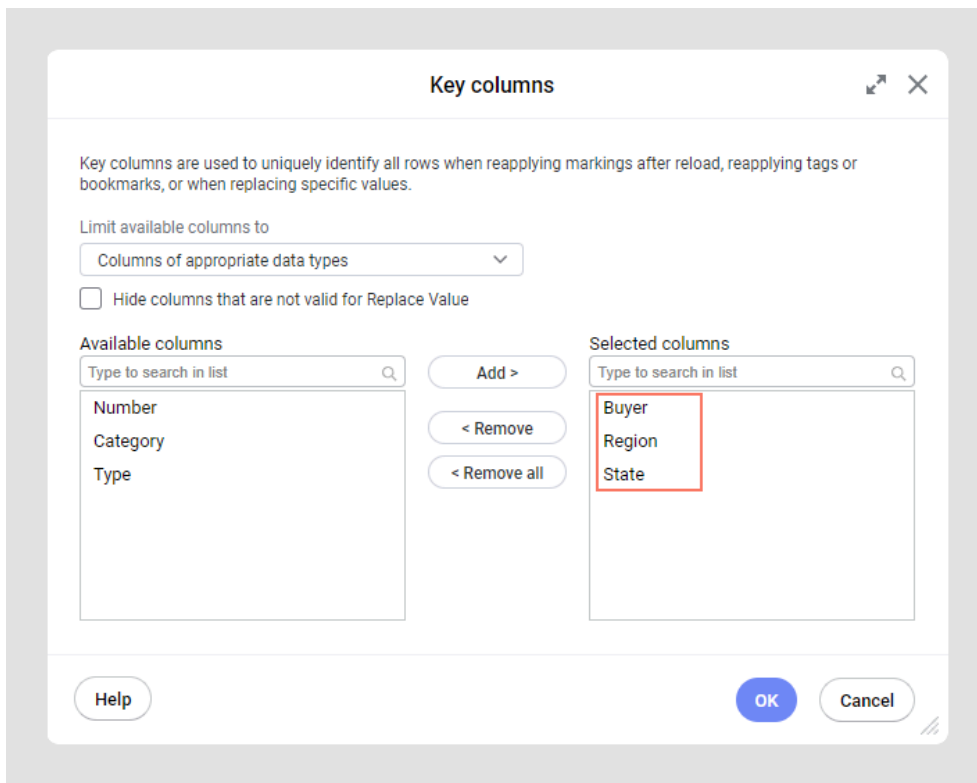
- **All columns**

This option shows all columns in the data table. You might have a data table where you can uniquely identify your rows using other type of columns (for example, a Date column).

You can also change the **Hide columns that are not suitable for Replace Value** check box, to find more or less columns.

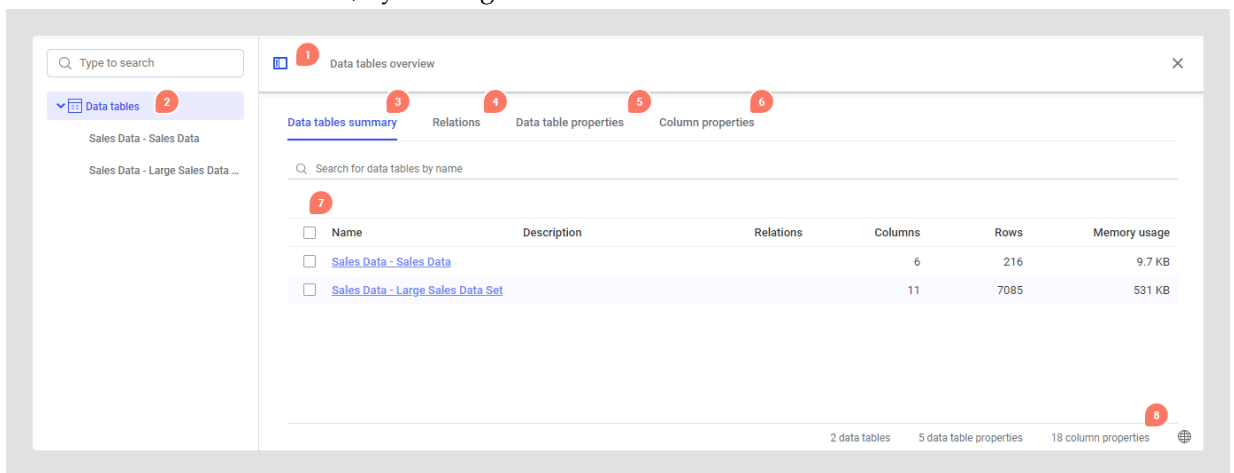
Some columns that are valid key columns for keeping markings, tags and bookmarks are not valid when using the 'Replace Specific Value'-transformation directly from a table visualization. This applies to any column that has been modified using a column conversion from the column view in the expanded data in analysis flyout (that is, when the column modification was done on the final data table). When this check box is selected, all columns that would result in invalid keys for Replace Value will be hidden.

In the image below, three columns have been added to uniquely identify each buyer in this data table.



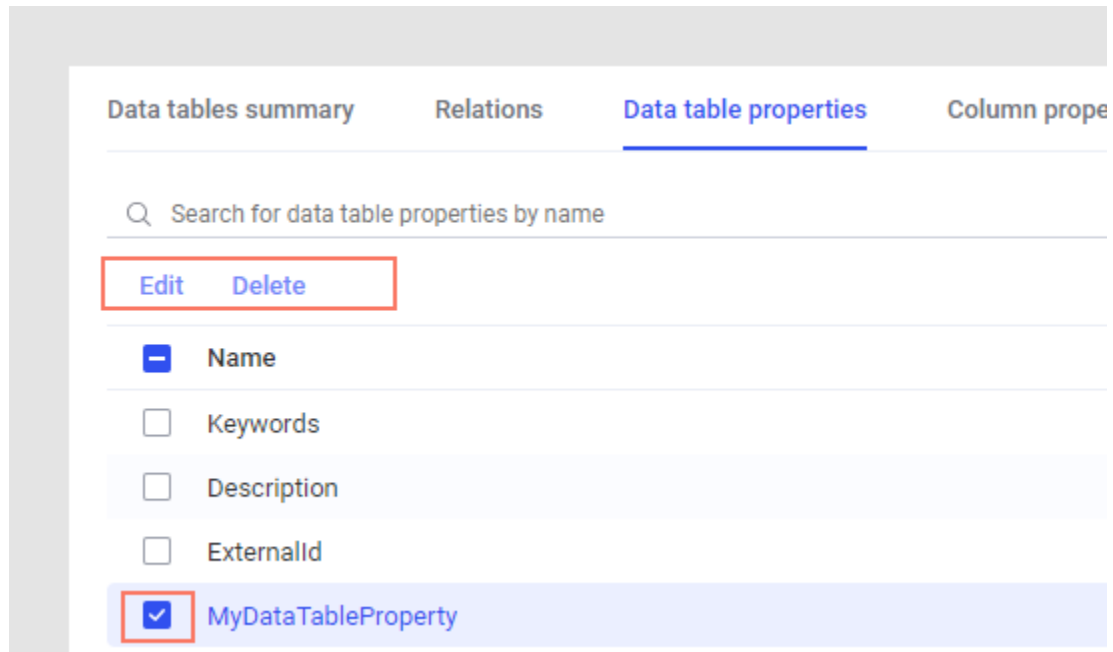
## Data tables overview

The data tables overview is a helpful tool if you have more than one data table in the analysis. You can reach it from the **Data canvas**, by clicking **Data tables** in the data canvas sidebar.



1. The data canvas sidebar can be shown or hidden by clicking on the icon at the top left part of the data canvas.
2. Click on the **Data tables** header to reach the **Data tables overview** which contains different tabs.
3. The **Data tables summary** shows all data tables in the analysis and lists metadata about each table.
4. The **Relations** overview shows [how the data tables are related](#) to each other and lets you add, edit and remove relations.

5. The **Data table properties** lists all [data table properties](#) and lets you [create new](#) properties and [edit the default values](#). To edit the value for a specific data table, see [Editing data table or column property values](#) on page 247.
6. The **Column properties** lists all [column properties](#) and lets you [create new](#) properties and [edit the default values](#). To edit the value for a specific data table, see [Editing data table or column property values](#) on page 247.
7. Select the check boxes to access operations applicable to the selected items, which will be shown as links above the table.



### Adding or editing a relation between two data tables


You can add relations to carry over markings and, optionally, also filtering between data tables in the analysis. A relation is created by defining how values should be matched using one or more identifier column.

Read more about relations in [Related data tables, joins and column matches](#) on page 74, and see also [Filtering in related data tables](#) on page 590.

#### Prerequisites

You must have at least two data tables in the analysis and the analysis must be in **Editing** mode.

#### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. In the data canvas sidebar, click on the **Data tables** header to reach the **Data tables overview**.
3. Click **Relations**.
4. Click **Add relation**.
5. Select the two data tables you want to connect from the **Data table** drop-down lists.
6. For each data table, select or type the **Column or expression** to use when matching the data tables. You might need to modify one or more of the columns you use to get a good match between the values in the different data tables. For example, if the identifiers are written in uppercase letters in one of the data tables and in lowercase letters in the other, you can use the expression

`Lower([MyColumn])` instead of just `[MyColumn]` with the uppercase column, and change the letters to lowercase. For data tables referring to cube data sources you can specify that the match should be based on keys rather than values by selecting the `KeyOf` method for the cube data table.


The **Sample** field shows the result of the selected expression after any specified functions have been applied and you can verify that values match as expected.

7. If needed, you can click **Add column pair** to add more rows with a column or expression to use in the match.

There is always only one relation between two data tables, but the relation can contain multiple matches. This means that the background (and API) expression describing the entire relation is on the form `f=g and h=i and ...`, where each column pair is separated by an "and".

8. When you are done, click **OK**.

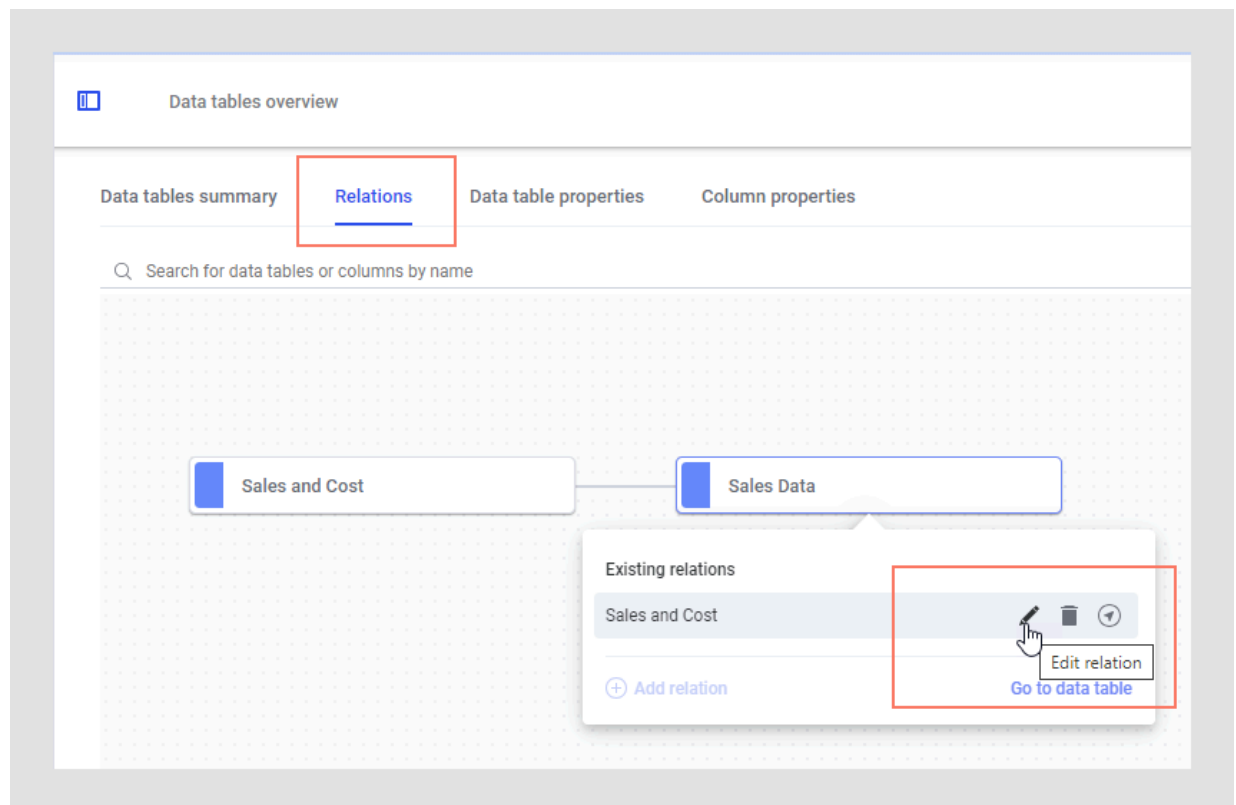


You can also add relations by navigating to a data table, selecting **Relations** , and clicking **Add relation**.

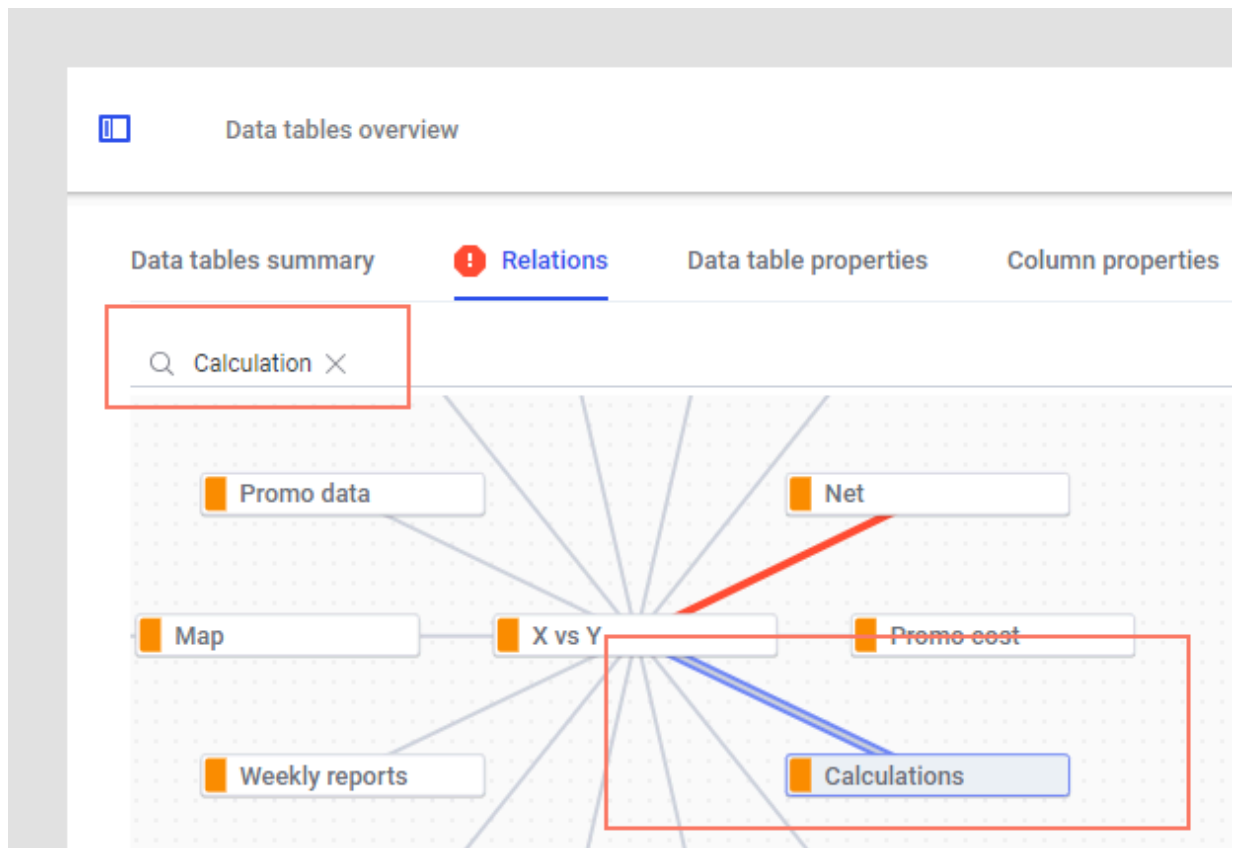
## Result

The relation is added to the analysis and you can see how the data tables are related in the **Relations** overview.

You can edit an existing relation by clicking on a node or an edge (the line between nodes), hovering with the mouse pointer on the row of interest, and then clicking **Edit relation** in the popover.



Any previously defined relations that contain errors are shown in red. You can also use the search field to search for a specific data table or column. Plain search matches the beginning of the data table or column name but you can also use wildcards (\*) to match parts of the names. All matches will be shown with a blue outline in the graph:



Use the scroll wheel on the mouse to zoom in or out in the relations graph, or drag the canvas to move the graph to a new position.



You can edit the color for a data table group by clicking on the color stripe to the left of the data table name on a node.

## Data tables in Spotfire

The data table is a central concept in Spotfire. All visualizations you create are based on data from one or more data tables in the analysis. You automatically get a data table when you add the first data to an analysis, but the data table can often be refined by adding more data from other sources (as rows or columns), by adding calculated columns or hierarchies, applying transformations, and so on. You can work with your data table in the data canvas. An analysis can contain one or more data tables and all data tables are saved when the analysis is saved.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client. Adding relations can be done using any client but adding or modifying column matches is done using the installed client.

The data in a data table can be either fetched from a data source, or created within the application. Data loaded from a data source can be handled either in-memory or in-database depending on how it is added to the analysis. See [Load methods](#) on page 58 for more information. In-memory data tables have one or more columns and zero or more rows, whereas in-database data tables technically do not contain any data but simply fetch the requested data directly from the source. See [Types of data in Spotfire](#) on page 45 for more information.

When there are multiple data tables in an analysis, they can either be completely separate from each other, or, they can be more or less tied to each other. See [Related data tables, joins and column matches](#) on page 74 for more information.



- **Relations:** When data tables are related, marking is carried over from one data table to another and filtering can be configured to be carried over, if desired.



You can specify whether or not filtering in a data table should affect what is shown in visualizations used by other, related data tables. The default setting is to ignore filtering in related data tables. See [Filtering in related data tables](#) on page 590 for more information.

- **Column matches:** Data tables can also have column matches between them, if columns are of the same data type. They are often created automatically, if the data is similar. A column match is used to aggregate data correctly when the data from different data tables is [combined in a single visualization](#). Column matches are much looser in nature than the relation mentioned above, because they do not join the data tables together. Marking and filtering is still individual for each data table, even if they have columns matching between them. To learn more, see [Column matches](#) on page 212. To see which column matches exist in the visualization, open the Column Matches tab in Data Table Properties in the installed client. You can also see which column matches are used in a specific visualization by opening the **Visualization properties** dialog and going to the **Data** section for the visualization of interest.


### On-demand data tables

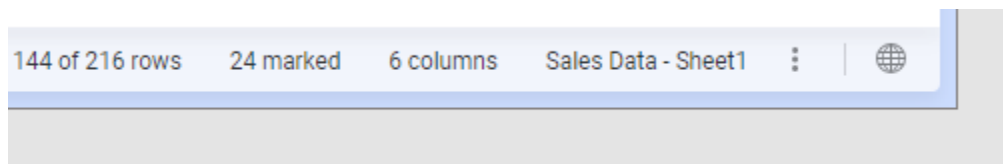
On-demand data tables are data tables to which only rows related to a defined input are loaded. The input could be something like the marked rows in another, related, data table, the filtered rows of another data table or a property value selected in a text area. Changing the input means changing the "demand", that is, that more, fewer or other rows are loaded into the data table. On-demand data tables can be used by details visualizations, and only data from information links or data connections can be loaded on demand.

### In-database data tables

Because data from in-database data tables is retrieved only when needed, the use of an in-database data table as a details visualization can also be seen as a type of on-demand visualization. For obvious reasons, this is not applicable for streaming data, because it is in fact updated continuously and not only when needed.

### Information about the active data table in the status bar

The status bar is located at the bottom or the top of the window and shows information about the main data table used by the active visualization (the visualization that was last clicked upon). If the information is not visible, click .



The information displayed is:

- the current number of filtered rows and the total number of rows in the main data table (but if the data table is loaded on demand, then the number of currently loaded rows is shown).
- the number of marked rows,
- the number of available columns,
- the name of the main data table in the active visualization.

Note that in-database data tables (external data) do not have access to the detailed information about rows and columns because the data shown in a visualization is aggregated by the external data source and not within Spotfire.

## Multiple data tables in one visualization

Sometimes, the data you want to analyze in Spotfire is located in different data tables. Working with visualizations combining data from multiple data tables is not very different from working with data from a single data table. You can choose the visualization that best suits your data, you can filter, mark, and drill down in your data. However, a couple of concepts are important when configuring and working with a visualization combining data from different data tables.



When multiple data tables are used in one visualization, the join of the two tables is done *after* aggregation has been performed. If you want to perform joining of tables *before* any aggregations, you should rather [add rows](#) from one table to the other, or configure [structural relations](#) between tables within a data connection.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

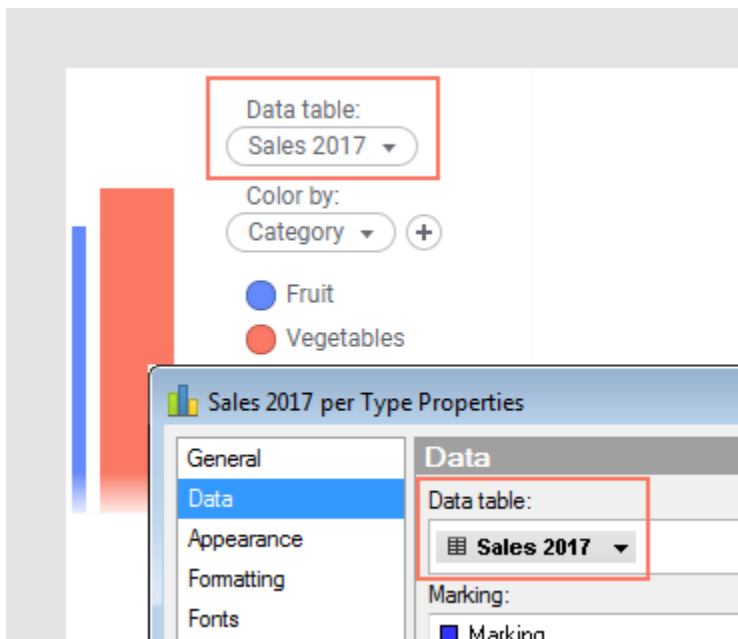
### The main data table

In a visualization combining data from different data tables, the main data table plays an important part. A visualization always has only one main data table, which is the anchor point in the data for the visualization. It defines what a row is in an unaggregated visualization, and the columns in the main data table are the columns that can be used to group the visualization in different ways. Consequently, the main data table columns control what becomes an item (such as a marker in a scatter plot, or a bar in a bar chart) in an aggregated visualization.

When you mark items in a visualization, details are shown for the columns in the main data table only. The main data table is also the data table that all expressions refer to by default, unless you explicitly specify that an expression should refer to another data table in the visualization by using the qualified column name, `[Data Table Name].[Column Name]`. (For example: `Sum([Sales Previous Year].[Sales])` where 'Sales Previous Year' is the data table name and 'Sales' is the name of the column.)

To get the most out of your data, think through which data table is best suited to be the main data table before you start configuring the visualization.

You select the main data table in the **Data** section of the **Visualization properties**, or in the data table selector in the legend. In the image below, the name of the main data table is 'Sales 2017':

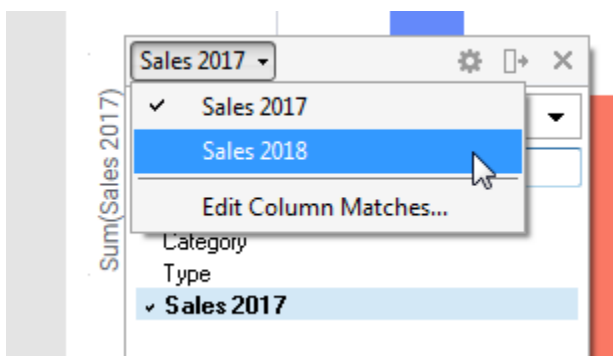


The **KPI chart** is slightly different from other visualizations. It can consist of several KPIs, where each KPI can be viewed upon as a visualization of its own. That is, each KPI has its own main data table.

### Additional data tables

Columns from other data tables than the main data table can be used on aggregating axes in the visualization, but not on axes that are grouping the visualization.

To add a column from another data table, you can use drag-and-drop from the **Data in analysis** flyout or the **Filters** panel, or select a column from the column selector. Open the column selector and switch to the data table of interest; the column selector will switch to show the columns in the selected data table instead. The data table selector is only visible if you have multiple data tables in the analysis and there are **column matches** available, see below. Column matches can only be edited using the installed client, but when they exist, you can add another data table to a visualization from the web client.



### Matching columns

Another important concept to know about is column matching. For a visualization to show data from many data tables, at least one column that you are going to use to group the visualization in some way, should match a corresponding column in the other data tables in the visualization. A column is matching if it contains the same kind of data. If columns contain values of the same data type AND have identical column names they will be matched automatically. For example, in the two data tables 'Sales 2017' and 'Sales 2018' below, the columns 'Category' and 'Type' match between the two data

tables. A basic rule when configuring a visualization is that all the categories you are going to use in the visualization should exist in all the data tables. That way, matching of columns is going to be easy. However, there are exceptions to that rule. To learn more about that, see the section about *Missing column matches* under [Column matches](#) on page 212.

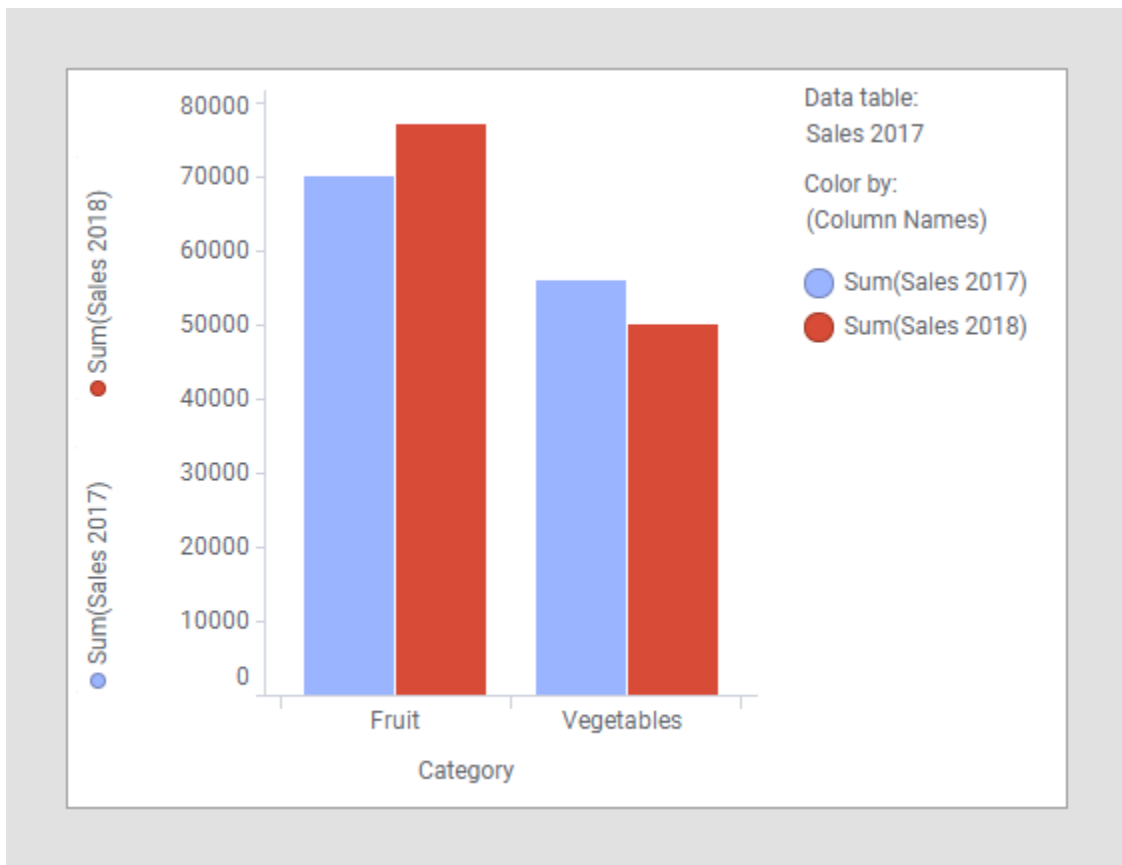
Sales 2017			Sales 2018		
Category	Type	Sales 2017	Category	Type	Sales 2018
Fruit	Apples	19000	Fruit	Apples	25000
Fruit	Pears	22000	Fruit	Pears	21000
Fruit	Bananas	29000	Fruit	Bananas	31000
Vegetables	Cucumber	12000	Vegetables	Cucumber	13000
Vegetables	Tomatoes	25000	Vegetables	Tomatoes	20000
Vegetables	Lettuce	19000	Vegetables	Lettuce	17000

If no automatic matches are found, you can add matches manually. To learn more about when and how to match columns manually, see [Column matches](#) on page 212, [Adding column matches manually](#) on page 216, and the following topics.

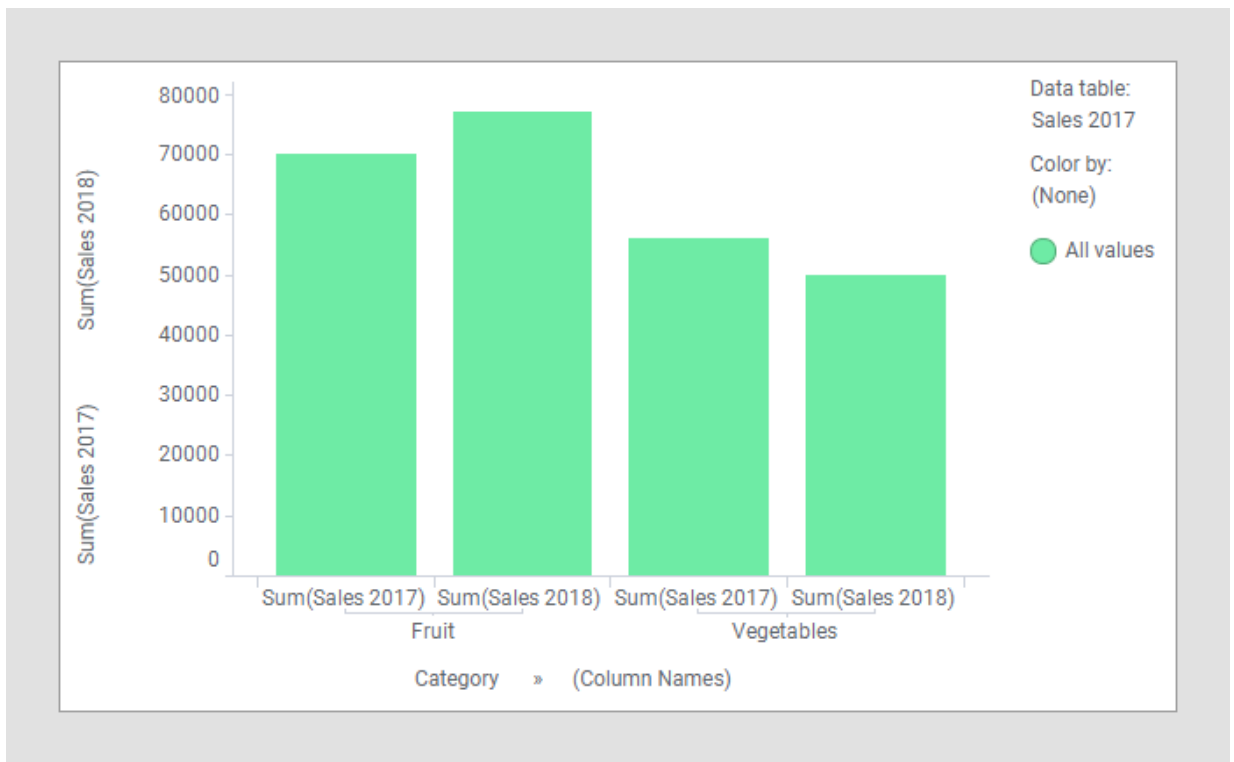
While it is generally not necessary to define a relation between data tables in addition to a column match, it can sometimes be useful to do so. With a relation between two data tables, marking and filtering from one data table can be propagated to the other data table. To read more about how to specify how filtering should work in related data tables, see [Filtering in related data tables](#) on page 590.

### Basic example

Comparing the data in the two data tables above in a bar chart does not require any special adjustments. Just load the two data tables into Spotfire, create the bar chart, select one of the categorical columns on the category axis, and then select the two columns 'Sales 2017' and 'Sales 2018' on the value axis.



Because the columns 'Category' and 'Type' in those two data tables have identical names and contain values of the same data type, they have already been matched automatically. As the example illustrates, the main data table is 'Sales 2017', and therefore, the column used on the category axis originates in that data table. In this example, any of the two data tables could be used as the main data table, because the categorical columns are the same in both data tables. As always, when multiple columns are used on the value axis of a bar chart, (Column Names) should be used to group the visualization. In the image above, the (Column Names) option is used to color by, but using it to trellis by or, as in the image below, adding it to the category axis are other possibilities.

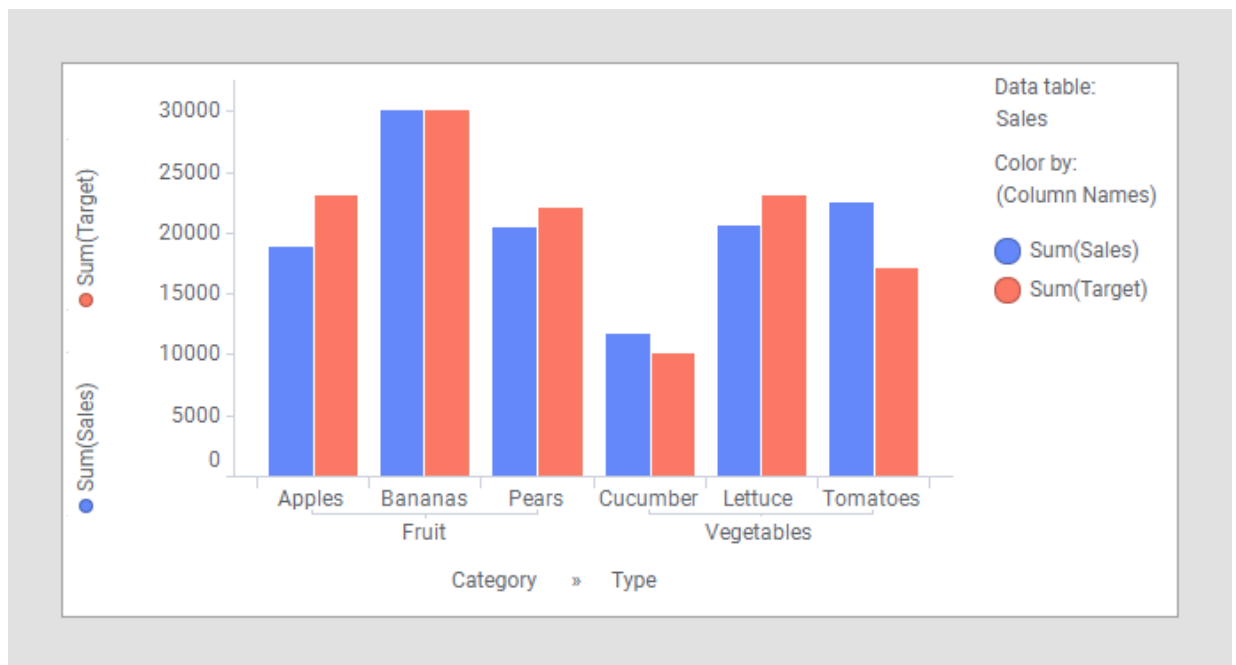


### Different levels of detail

In the example above, the data tables had more or less the same columns; 'Category', 'Type', and a column containing the sales figures. You can also compare data from data tables with data on different levels of detail in a visualization. For example, you might want to compare sales targets for a certain year with the actual sales so far for the year. Perhaps you have a data table containing sales targets for fruits and vegetables. For each fruit and vegetable type, one single row represents the target, as seen in the 'Targets' data table below. In another data table, you might have the actual sales data for the current year, as seen in the 'Sales' data table below. In this data table, each sales transaction is represented by one row, which means that for each type of fruit and vegetable, there are several rows of sales figures.

Targets			Sales			
Category	Type	Target	Transaction	Category	Type	Sales
Fruit	Apples	23000	FA-001	Fruit	Apples	1840
Fruit	Pears	22000	FA-002	Fruit	Apples	1310
Fruit	Bananas	30000	FA-003	Fruit	Apples	1290
Vegetables	Cucumber	10000	FA-004	Fruit	Apples	1780
Vegetables	Tomatoes	17000	FA-005	Fruit	Apples	2000
Vegetables	Lettuce	23000	FA-006	Fruit	Apples	790
			FA-007	Fruit	Apples	1550
			FA-008	Fruit	Apples	1520
			FA-009	Fruit	Apples	1280
			FA-010	Fruit	Apples	1670
			FA-011	Fruit	Apples	1960
			FA-012	Fruit	Apples	1770
			FB-001	Fruit	Bananas	2590
			FB-002	Fruit	Bananas	1550

By combining data from those two data tables in a bar chart you can see which fruits and vegetables have reached their targets this year:



### Recommended workflow

If you are unsure how to configure a visualization combining columns from different data tables, this recommended workflow can be helpful.

#### 1. Choose the main data table

Start by having a look at the data in the different data tables, and try to answer a couple of questions. What data do they contain? What do you want to visualize based on that data? A data table containing categories you would like to group your visualization by is a good candidate for the main data table. For instance, you might want to group by region, department, salesperson, product type, or similar.

## 2. Configure the visualization with only the main data table

Add a visualization of the type you want to use, and then configure as much as you can of that visualization with columns only from the main data table. Select how and by which columns the visualization should be grouped, and if the main data table also contains columns that you want to show as aggregated, add those columns to the appropriate axes as well.

## 3. Add the aggregating measures

When the visualization has been configured as much as possible with only main data table, you can start adding aggregated columns from other data tables.

### Column matches

When you combine data from different data tables in one visualization, you must consider how the columns in the data tables match. A rule of thumb is that all the columns you are planning to group the visualization by, should exist in all the data tables. For example, this could be columns that will define what a marker is in a scatter plot, columns you wish to set on the category axis of a bar chart, or columns you wish to trellis or color the visualization by. If your data is structured that way, configuring the visualization and matching columns will be easy. However, there are cases when it is OK that some columns in the main data table do not have matches in all the data tables. And, even if the data is configured the recommended way, you might sometimes need to make a few manual adjustments.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

### Automatic column matching

When adding multiple data tables to an analysis (in any Spotfire client), Spotfire will automatically match columns with the same name and data type. For example, if you load the data tables below into Spotfire, a match will automatically be added between the columns named 'Product':

Product	Sales 2017	Product	Sales 2018
Apples	19000	Apples	25000
Pears	22000	Pears	21000
Bananas	29000	Bananas	31000
Cucumber	12000	Cucumber	13000
Tomatoes	25000	Tomatoes	20000
Lettuce	19000	Lettuce	17000

When the column match is available you can use the data from both tables in a single visualization, as described in [Multiple data tables in one visualization](#) on page 206.



One way to fix some issues with missing column matches is to rename the columns to group by in the additional data tables, so they use identical names (and data types) as in the main data table. Then, the automatic column matching will kick in.



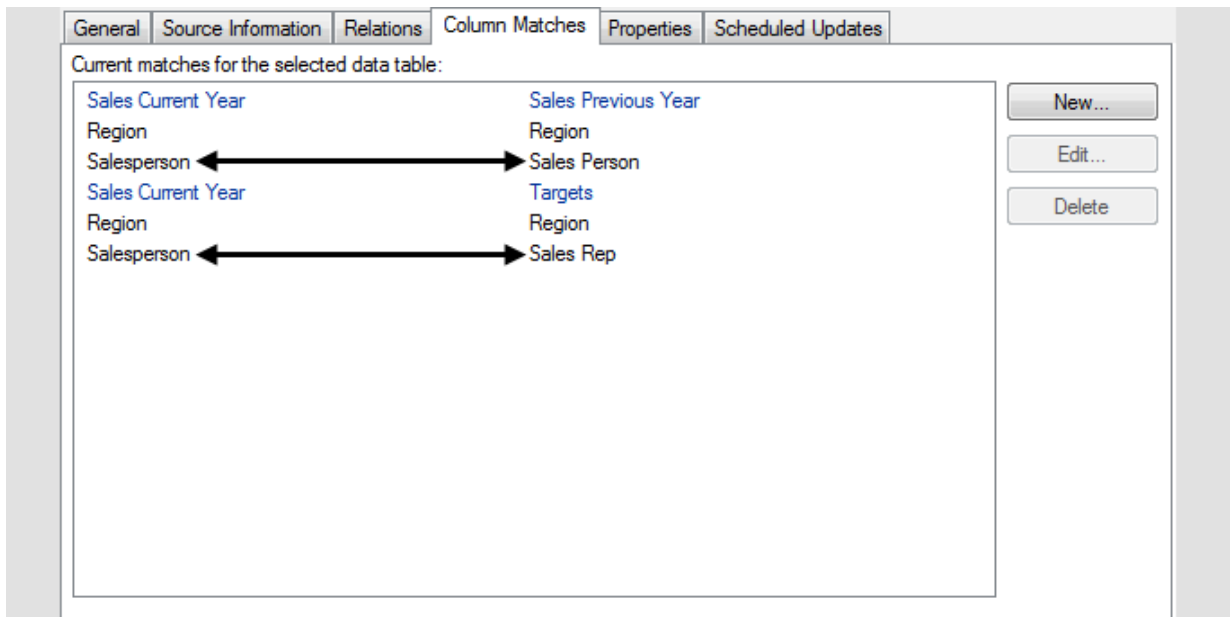
## Manual column matching

If the data tables that you want to use do not contain an exact column match directly, you can add and edit column matches using the installed client. This can be necessary if the columns you want to match have different names, or data types, or if the values in the columns use different casing.

For details, see [Adding column matches manually](#) on page 216, and the examples following that topic.

## Viewing the current column matches

When matches have been added, they are shown in the **Column Matches** tab of the **Data Table Properties** dialog (installed client only). Click on a data table to see the current matches for that data table:



The data table names are displayed in blue text, while the column names are displayed in black text. The selected data table is always shown on the left-hand side.

You can also view columns matched in a specific visualization in the **Data** section of the **Visualization properties**:



If matches are missing, this is indicated here, and you can add a match if necessary. If a column has more than one match to a column in another data table, you can select which match to use in this specific visualization.

## Missing column matches

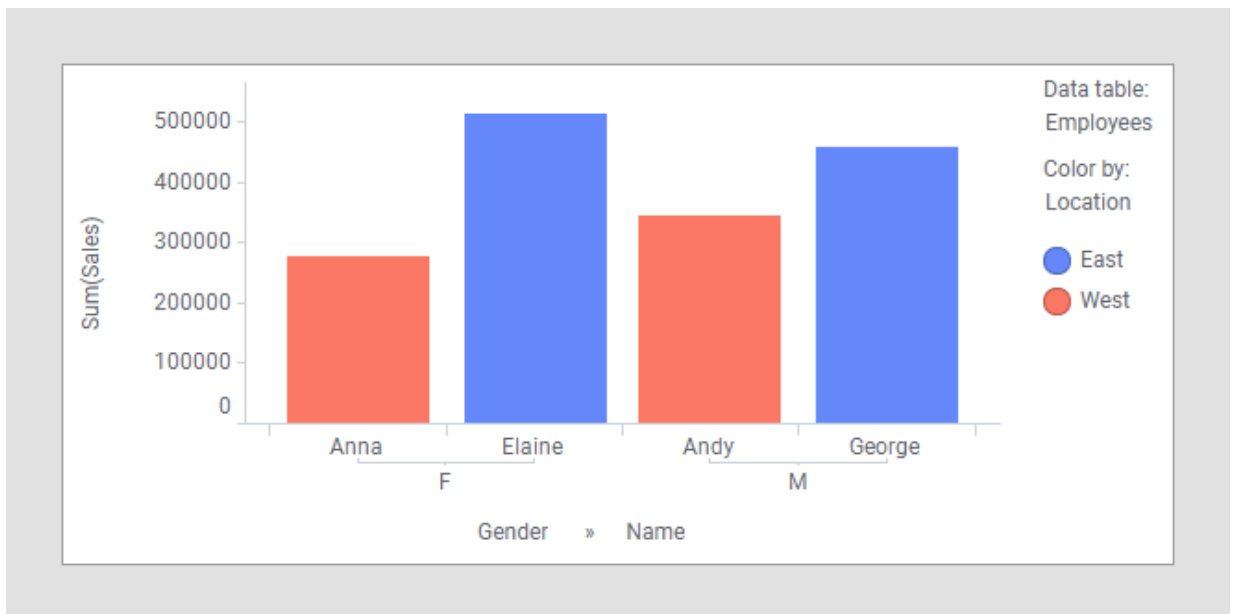
As mentioned above, a rule of thumb is to make sure that all the columns you want to group the visualization by exist in all the data tables that will be used in the visualization. But there are cases when the visualization will be valid even if columns that are used on a grouping axis do not have matches to columns in the other data tables. The examples below illustrate this.

## The visualization is valid

If you know that a category is unique, you can use it on a grouping axis even if the column only exists in the main data table. The left-most data table below contains information about employees in a sales organization: their names, gender, and office location. The right-most data table contains the total sales for each person.

Employees			Sales	
Name	Location	Gender	Name	Sales
Andy	West	M	Andy	343000
Anna	West	F	Anna	275000
George	East	M	George	456000
Elaine	East	F	Elaine	512500

The only possible column match for the data tables is 'Name', because that is the only column that exists in both data tables. The 'Employees' data table contains only categories, while the 'Sales' data table contains the sales figures you probably want to compare. Combining these two data tables would make it possible to look at the data from location and gender perspective as well as just comparing the sales figures per person. Because each person in this example identifies only as one gender and works only at one office location, you can actually group by all the columns in that data table, provided that you select 'Employees' as the main data table, as seen in the bar chart below:



'Gender' and 'Name' define the category axis, while 'Location' defines the colors of the bars.

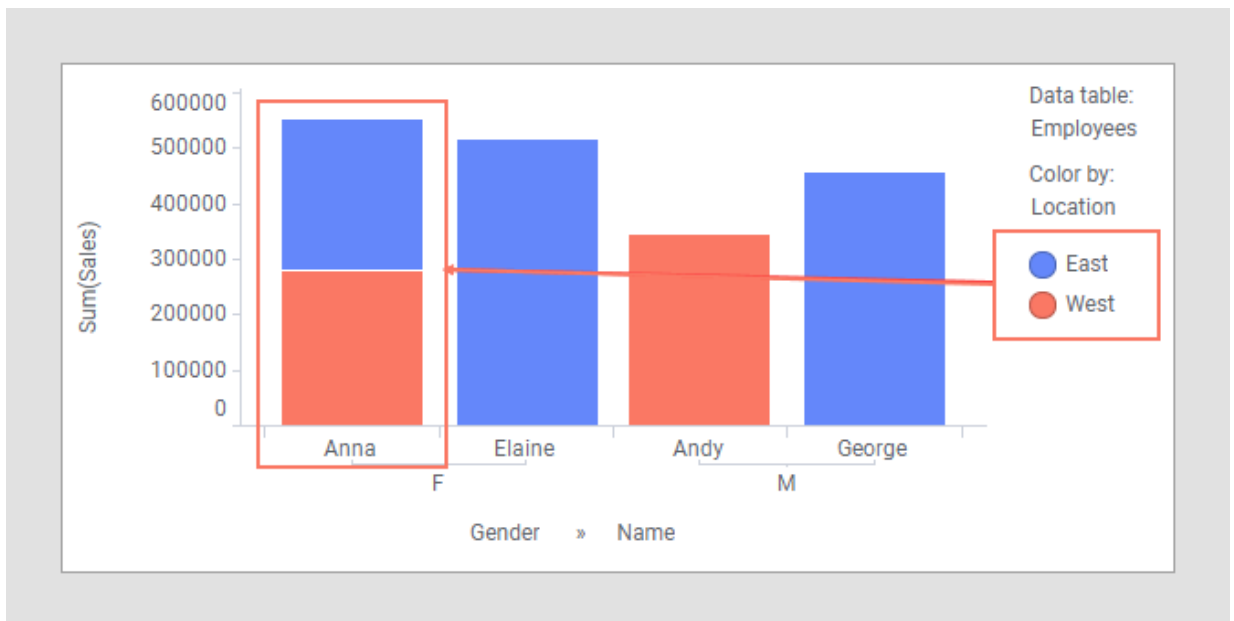
If you are going to group a visualization by unmatched columns, it is important to make sure the categories are unique. If the categories are not unique, you will end up with a visualization that shows incorrect data, as seen in the next example.

#### The visualization is NOT valid

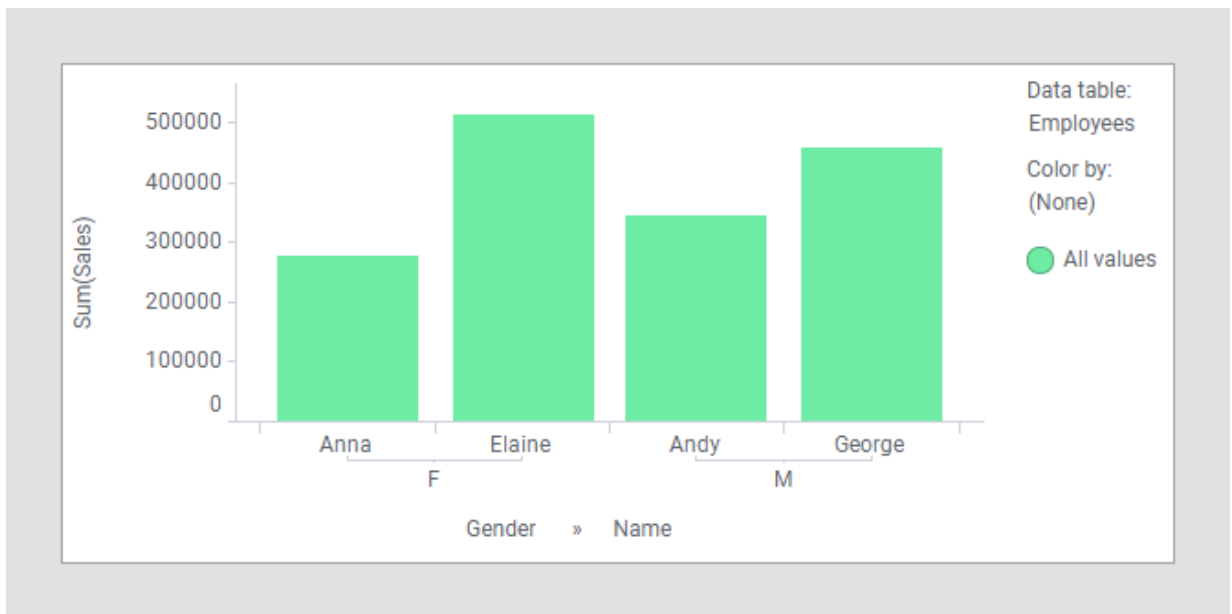
The data tables below are similar to the ones in the previous example, but the 'Employees' data table has been updated to reflect that Anna is actually working out of both offices. The 'Sales' data table does not have to be updated due to this change, because the amount Anna sold for last year has not changed, and has nothing to do with which offices she works out of.

Employees			Sales	
Name	Location	Gender	Name	Sales
Andy	West	M	Andy	343000
Anna	West	F	Anna	275000
Anna	East	F	George	456000
George	East	M	Elaine	512500
Elaine	East	F		

However, this change in the main data table will affect the visualization. Because the bars are colored by 'Location', and Anna belongs to both 'East' and 'West', she will now appear twice; once for the East office, and once for the West office. This means that it now looks like Anna has sold for twice the amount she actually has, which is of course incorrect.



With Anna working out of two offices, 'Location' can no longer be used on a grouping axis in the visualization. However, by removing 'Location' from the color axis, you can turn the bar chart into a valid visualization again:



Now, each person is shown only once in the visualization, and while you cannot group by location, you can still group by gender.

If you are certain that you have a valid visualization, you can switch off warnings about missing column matches as described in [Hiding warnings about possible mismatches](#) on page 222.

### Adding column matches manually

If you want to show data from multiple data tables in a single visualization, the data tables must have at least one column match available. If the data in the data tables is similar, column matching often happens automatically, but you can add manual matches using the installed client when needed.

Read more in [Multiple data tables in one visualization](#) on page 206 and [Column matches](#) on page 212.



Another way to fix some issues with missing column matches is to rename the columns to group by in the additional data tables, so they use identical names (and data types) as in the main data table. Then, the automatic column matching will kick in.

#### Prerequisites

Manual column matches must be authored in the installed client.

#### Procedure

1. In the installed Spotfire client, open the analysis of interest or load data tables into a new analysis.
2. On the menu bar, select **Data > Data Table Properties**, and open the **Column Matches** tab.
3. In the **Data tables** list, select the data table to use as the main data table in the visualization.
4. Click **New**, to the right of the list showing the **Current matches for the selected data table**. The **New Match** dialog is opened.
5. In the **Right data table** drop-down list, select the data table to add a match for. The data table in the **Left data table** drop-down list is already set to the data table that was selected in the Data Table Properties dialog.

6. In the **Left matching column** selector, choose the column to use for the match from the left data table.



If needed, click the arrow on the column selector and the arrow on **Methods** in the drop-down list and choose a method to apply to the column to get a better match. For example, you can change the data type for the column.

7. In the **Right matching column** selector, choose the column to use for the match from the right data table.



If needed, you can add a method to the column. See step 6.

8. If needed, you can also add a **Left column value transformation** and/or a **Right column value transformation** by clicking the arrow on **Value** and selecting something under **Methods**.

In this step, you can transform the values within the column to get a better match. For example, you can use the "Upper" method to convert all values in a column to use uppercase letters, if the casing is different in the two columns and you therefore not get any matches. The available methods differ with the data type of the selected column. See [Creating matches to ensure same grouping of values](#) on page 220 for an example.

9. Click **OK** to close the dialog.
10. Repeat steps 4-9 if more column matches are needed.
11. Click **OK** to close the Data Table Properties dialog.

## Result

The column matches are added to the selected data tables.

## Data tables with matching columns using different names

If you want to show data from multiple data tables in a single visualization, the data tables must have at least one column match available. This example shows how you can add a manual column match when columns to match exist but they have different names.

If you load the three data tables below into Spotfire, matches will automatically be added only between the columns named 'Region' because that is the only column with an identical name in all of the data tables.

Sales Previous Year			Sales Current Year			Targets		
Sales Person	Sales Prev Yr	Region	Salesperson	Sales Current Yr	Region	Sales Rep	Target	Region
Andy	306000	West	Andy	275000	West	Andy	300000	West
Anna	262000	West	Anna	343000	West	Anna	300000	West
Elaine	549000	East	Elaine	512500	East	Elaine	550000	East
George	423000	East	George	456000	East	George	430000	East

However, if you want to compare the performance of the four salespersons rather than sales per region, you must use a column containing the names of the persons on an axis that groups the visualization in some way; for instance, on the category axis of a bar chart. In this example, those columns are named 'Sales Person', 'Salesperson', and 'Sales Rep' respectively, which means they are not identical and have not been matched automatically. But the columns contain the same values, so they can be used to group an axis in the visualization. To do so, the column matches must be added manually.

To use the data table named 'Sales Current Year' as the main data table, and to include data from the other two data tables in the same visualization, you must add two matches: 'Salesperson' - 'Sales Person'

and 'Salesperson' - 'Sales Rep'. As long as you use 'Sales Current Year' as the main data table, it is not necessary to add a match between 'Sales Person' and 'Sales Rep', but you can add a match between them as well.

## Prerequisites

Manual column matches must be authored in the installed client.

## Procedure

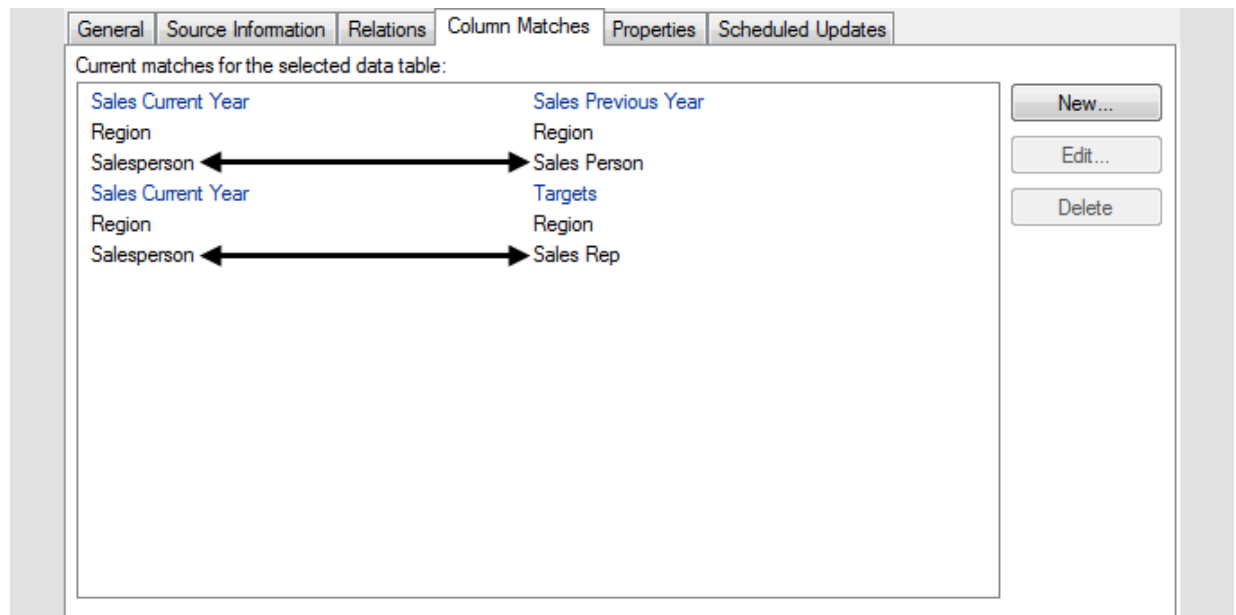
1. In the installed Spotfire client, load the data tables, select **Data > Data Table Properties**, and open the **Column Matches** tab.
2. In the **Data tables** list, click to select the main data table: 'Sales Current Year'.
3. Click **New**, to the right of the list showing the **Current matches for the selected data table**.
4. In the New Match dialog, in the **Right data table** drop-down list, select the 'Sales Previous Year' data table.

The data table in the **Left data table** drop-down list is already set to 'Sales Current Year' because this data table was selected in the Data Table Properties dialog.

5. In the **Left matching column** selector, choose 'Salesperson'.
6. In the **Right matching column** selector, choose 'Sales Person' and click **OK**.
7. Repeat the steps to add the second column match, but select the 'Targets' as the **Right data table**, and 'Sales Rep' as the **Right matching column** instead.
8. Click **OK** to close the Data Table Properties dialog.

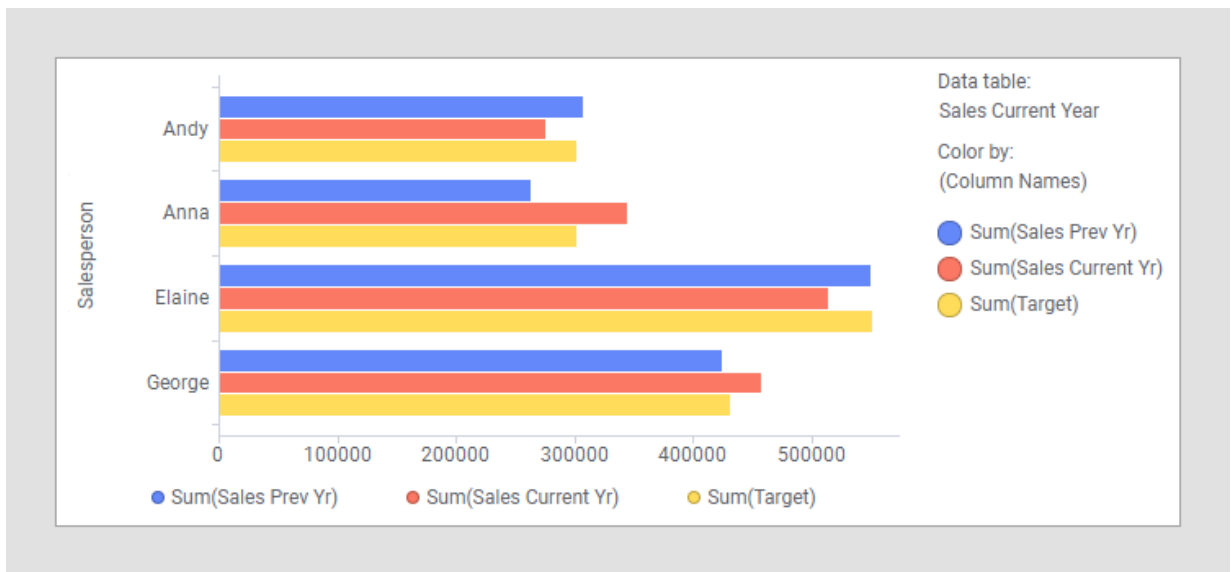
## Result

When matches have been added, they are shown in the **Column Matches** tab:



Note that column matches for the column named 'Region' had already been added automatically.

With the added matches, you can now create a visualization where 'Sales Current Year' is the main data table, and the column 'Salesperson' is used on a grouping axis:



### Editing matches to transform values with different casing

Perhaps you have two data tables which both contain a string column named 'Product', but in one data table, the values are written in lowercase ('apples'), while in the other data table they are written with the first letter in uppercase ('Apples'). Because the columns have the same name and data type they will be matched automatically. However, to be able to combine data from these columns in one visualization, the values in one column must be transformed so that they are written in the same way in both columns.

By adding a transformation step to the column match rather than converting the actual column values you can use the original value names in the visualization.

#### Prerequisites

Manual column matches must be authored in the installed client.

#### Procedure

1. In the installed Spotfire client, open the analysis of interest or load data tables into a new analysis.
2. On the menu bar, select **Data > Data Table Properties**, and open the **Column Matches** tab.
3. In the **Data tables** list, select the data table to use as the main data table in the visualization.
4. Select the match between the two columns named 'Product' in the **Current matches for the selected data table** list.
5. Click **Edit**.
6. In the Edit Match dialog, make sure the columns named 'Product' are selected in the **Left matching column** and **Right matching column** selectors.

The **Sample value** under the left and right matching column shows the first value in the selected column (provided it is not in-db data).

<b>Left matching column:</b> <div>Product ▼</div>	<b>Right matching column:</b> <div>Product ▼</div>
<b>Sample value:</b> <div>Apples</div>	<b>Sample value:</b> <div>apples</div>

What you select here is what will be seen on the axes in a visualization.

7. Open the column selector for the column containing values in uppercase (for example, under **Left column value transformation**), and select **Methods** to open the pop-up menu.

The transformations are always applied on the value that comes out after any methods have been applied in the column matching step above. This is why the Left and Right column transformation lists show the name 'Value' rather than a column name.

8. Select **Lower**.

In the **Sample value** fields you can now see that the values are written as 'apples' in both columns.

<p>Left column value transformation:</p> <p>Lower(Value) ▾</p> <p>Sample value:</p> <p>apples</p>	<p>Right column value transformation:</p> <p>Value ▾</p> <p>Sample value:</p> <p>apples</p>
---	---

The transformations are transparent to what is shown in the visualizations, which means that even if you apply the transformation Lower(Value) you will not have to see this in the visualizations, but can use 'Apples' with a capital A on the axes.

9. Optionally, click **Preview** to see a preview of the first 100 distinct values and make sure that all values match.  
The preview is the only way you can control whether values from in-database sources match, because automatic samples are disabled for in-db data to keep the stress on the external system to a minimum.
10. Click **OK** to close the dialog.
11. Click **OK** to close the Data Table Properties dialog.

## Result

The values in the columns are now in the same format and can be combined in one visualization.

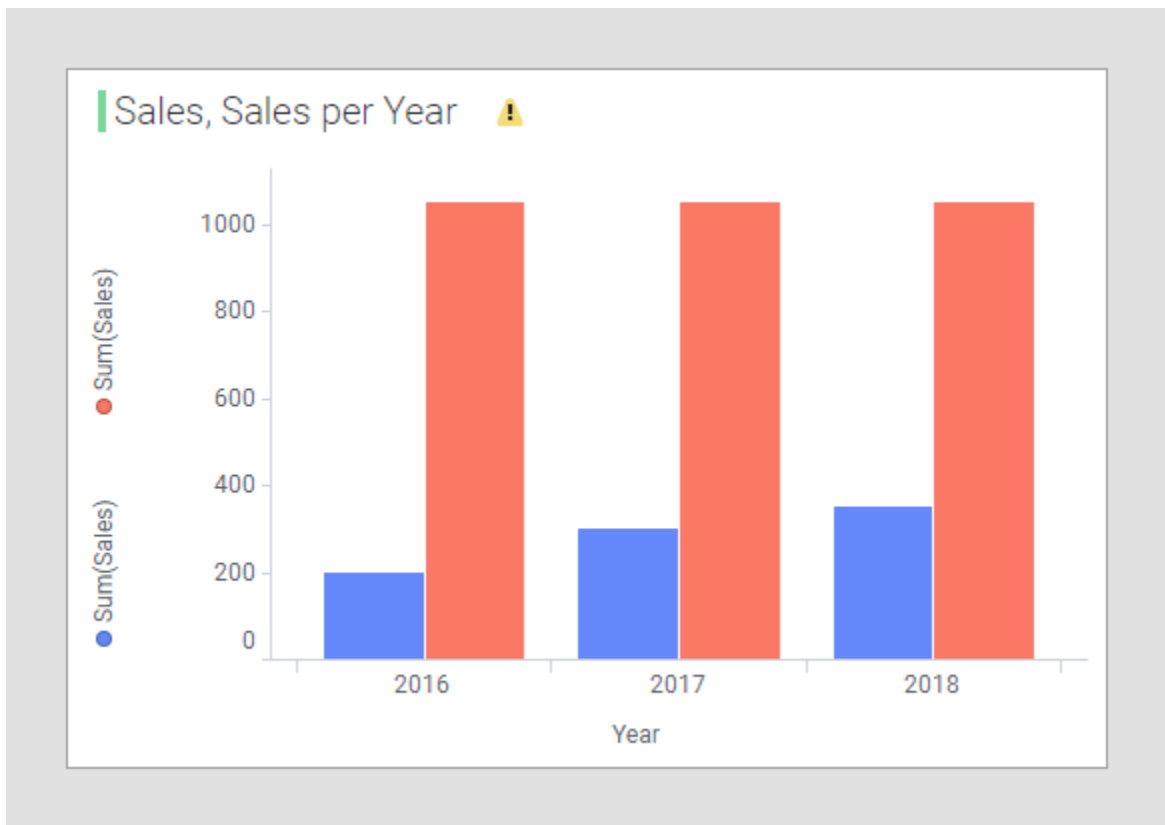
The transformation step is always done in-memory, independent of the original source of the data, which makes it possible to perform changes that the original data source does not support. For example, some in-database sources do not allow you to switch data type of a column, but because the transformation step is done in-memory you can create matches with in-database data of different data types this way. It is also useful for in-database data where there are no row methods in the system (for example, OLAP cubes).

However, there are also cases when you must apply methods on the columns directly, rather than using a transformation, for example, when the grouping of values is affected by the result from the method applied. For cube data you might also want to specify that the match should be done by keys rather than by values and this is done by selecting the `keyOf` method on the matching column of the cube directly.

## Creating matches to ensure same grouping of values

If you have two columns where date or time values are in different formats you might need to apply a method to ensure the same grouping of values is used in both tables. For example, in one data table there might be an integer column named 'Year', with values such as 2016, 2017, and so on, and in another data table there might be a column named 'Transaction Date', which contains values in the format `DateTime`.





### Prerequisites

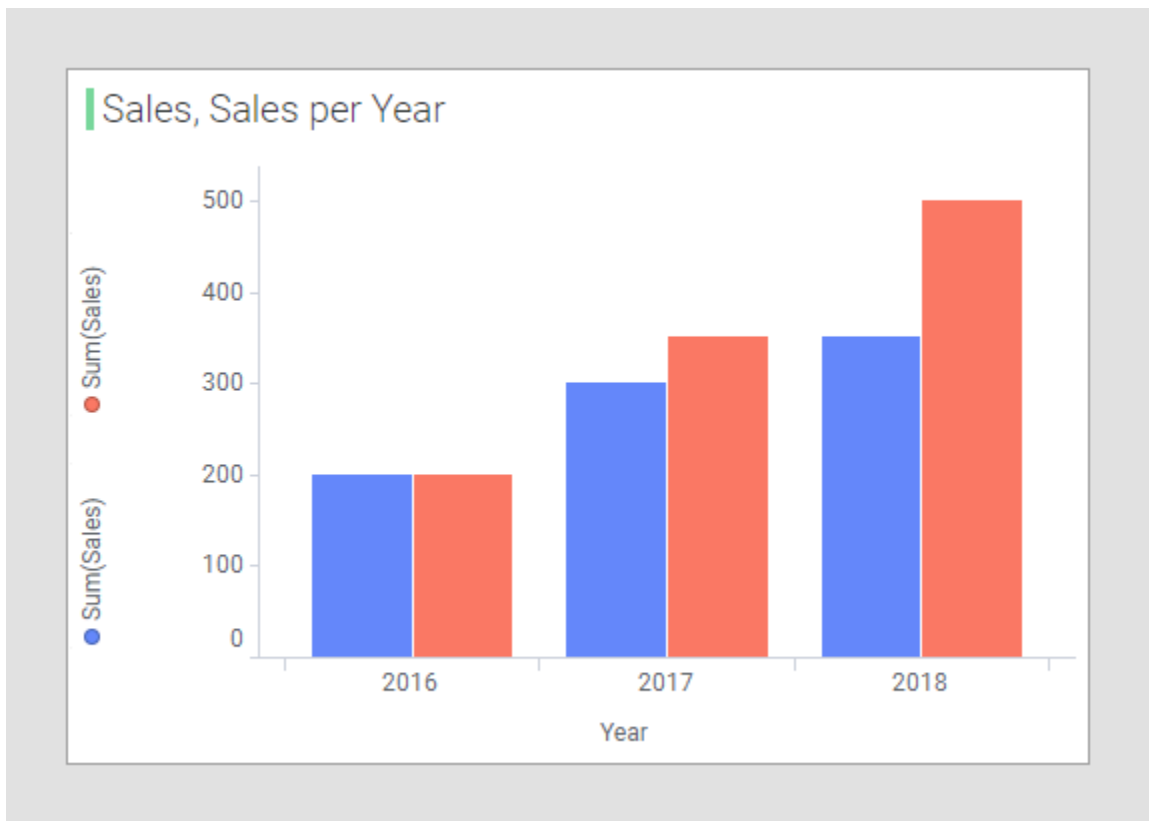
Manual column matches must be authored in the installed client.

### Procedure

1. In the installed Spotfire client, open the analysis of interest.
2. On the menu bar, select **Data > Data Table Properties**, and open the **Column Matches** tab.
3. In the **Data tables** list, select one of the data tables of interest.
4. Click **New**.
5. In the New Match dialog, select the other data table under **Right data table**.
6. Make sure the columns with the time-based information are selected in the **Left matching column** and **Right matching column** selectors.
7. Click the column selector for the `DateTime` column and click **Methods** to open the pop-up menu.
8. Select **Year**.  
To add a match between these columns, you must use the method **Year** on the 'Transaction Date' column. This will ensure that the values in the two columns are of the same format; 2016, 2017, and so on.
9. Click **OK**.

### Result

Instead of showing the smeared total values for the Transaction Date column, the visualization now shows the sum of sales for each year for this data table as well.

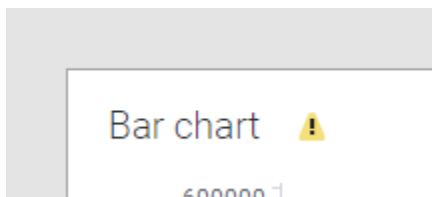


### Hiding warnings about possible mismatches

When working with multiple data tables in one visualization, it is important to make sure no categories are repeated if you group the visualization by a column that is only located in the main data table. Therefore, Spotfire will display a warning if you group by columns without column matches in the other data tables.

For general information and some examples of column mismatch, see [Column matches](#) on page 212.

The warning about possible mismatches is shown in the title bar of the visualization:



You can click the icon to see the whole message, which includes a list of all columns without matches.



In the installed client, if a warning is shown, you can open the message and click the link **Review matches** to come directly to the **Data** section in the **Visualization properties**. Here, you can review the current matches used in the visualization, and if necessary, you can add new matches.

When you have made sure the visualization does not show repeated categories, you can if you like, switch off future notifications about missing matches in the visualization.



If you hide warnings, it is important to make sure that no future changes of the visualization configuration introduce repeated values, because from then on, you will not be notified about this (unless you switch notifications back on again).

## Procedure

1. Open the **Visualization properties** and go to the **Data** section.
2. Locate the **Data table matching** section.
3. Clear the check box **Show notifications about mismatches**.

## Result

Notifications will no longer be shown in the visualization.

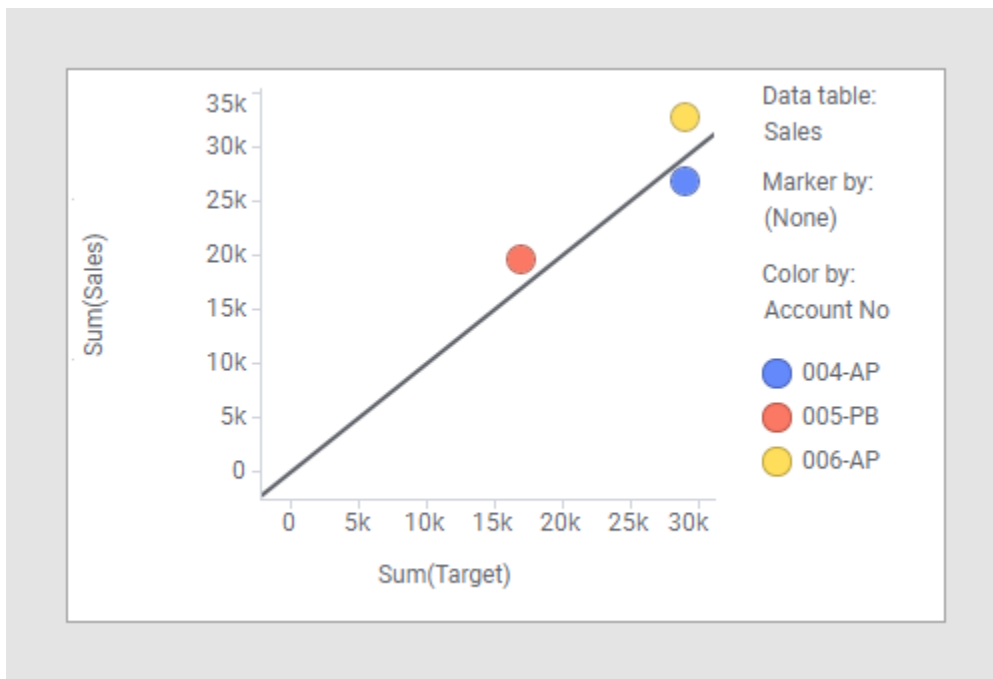
## Using labels to show data from other data tables

When column matches are available, you can sometimes choose to add labels from another data table to a visualization. This functionality is not available for all visualization types.

Perhaps you have the following three data tables:

Sales			Targets			Managers	
Account No	Sales	Year	Target	Product	Account No	Account Mgr	Account No
004-AP	8500	2016	17000	Apples	004-AP	Elaine	005-PB
004-AP	9500	2017	12000	Pears	004-AP	Elaine	006-AP
004-AP	8700	2018	5000	Bananas	005-PB	George	003-PB
005-PB	8500	2016	12000	Pears	005-PB	George	004-AP
005-PB	11000	2018	13000	Apples	006-AP	Andy	007-PB
006-AP	9000	2016	16000	Pears	006-AP		
006-AP	15000	2017					
006-AP	8700	2018					

And a visualization has already been created with the 'Sales' data table as the main data table:



The X-axis shows sum of target, which is data that originates in the 'Targets' data table. The Y-axis shows sum of sales, which is data that comes from the 'Sales' data table (the main data table). The markers represent the different accounts, which is also data from the main data table. The visualization shows how each of the accounts have performed compared to their targets, but it might also be interesting to see which person is responsible for which account. But in the main data table there is no information about the account manager so it is not possible to group by account manager. However, the data table named 'Managers' contains information about which person is manager for which account, and it is possible to get that information into the visualization by using labels as aggregated values rather than grouping categories.

### Procedure

1. Open **Visualization properties** and go to the **Labels** section.
2. Open the **Label by** column selector.
3. Select the 'Managers' data table and then the 'Account Mgr' column.
4. Select **UniqueConcatenate** as aggregation method.
5. Choose **Show labels for > All markers**.

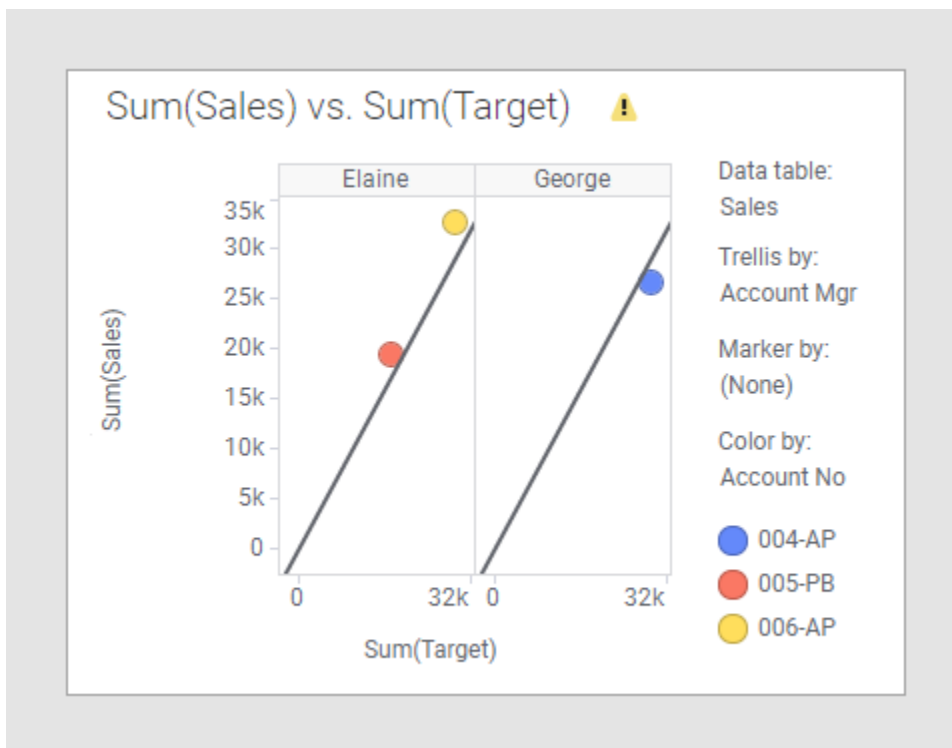
### Result

The scatter plot now shows labels with the account manager name for each of the accounts, as seen in the image below.



If you cannot add data from another data table as aggregated values, as in the example above, you can often add a column from another data table to the main data table instead. That way you can use the column to group the visualization by. To add a column from an already existing data table, go to the **Files and data** flyout, click **Other > Linked copy to data table in analysis**, and select the data table of interest. In the summary view, make sure that the option **Add as columns to** is selected, and that the main data table is selected as the data table to add data to.

If the 'Account Mgr' column from the example here is added to the main data table, you can group the visualization by that column, as seen in the visualization below:



Note that you will get a warning in the visualization because the 'Account Mgr' column does not have a match in the 'Targets' data table. In this case it is actually not a problem, and you could switch the warning off. Read more about when and how you can disregard this warning in [Column matches](#) on page 212 and [Hiding warnings about possible mismatches](#) on page 222.

## Replacing data

---

There are several different methods you can use to replace data in Spotfire. If linking the data to the source is not enough, you can select to replace a data source within a data table, or, to replace the entire data table. You can also replace the values in a column using a transformation, or a single value within a column using one of the Replace Value-options.

### Reload a linked source

The easiest form of replacing your data is to use linked data in your analysis. Linked data can be configured to always use new data when the analysis is opened, and you can reload it from the [data canvas](#) at any time. This is, most of the times, the preferred option when you want fresh data from the source at all times.

### Replace data table

If you select to [replace a data table](#), the operation will replace everything, including all sources within that data table. All previously added transformations, added columns or rows from other sources, deleted rows, or replaced values, will be lost. Calculated columns and inserted hierarchies will remain, though, because they are applied on the final data table.

Use this option to switch to completely different data, or to switch the type of data table from an in-db table to an imported table, or vice versa, which is not possible when replacing the data source only.

### Replace data source

Replacing a [data source](#) only replaces that particular data source, but keeps the other parts of the data table intact. That means that all transformations, calculations, added rows or columns, and operations that were used to build the final data table (all steps between the first added data source and the final data table in the source view of the data canvas) are kept as is.

Use this option to switch from a test database to a production database, or when your data remains exactly the same but is moved from one type of data source to another.

### Calculate and replace column (installed client only)

Use the **Calculate and replace column** [transformation](#) to modify a column using an expression before you use it in your analysis. For example, to remove a part of a string that is not needed.

### Replace value

Use [Replace value](#) to change a value on a specific row, or a misspelled value in all rows of a column, or use the functionality to [substitute empty values](#) in a column.

## Reloading data

If you know that the data sources used in your analysis are refreshed often and you want to make sure you view the absolutely latest data, you can reload data during a session, without having to reopen the analysis.

Use the following steps to reload most data tables in your analysis simultaneously. If you only want to reload a particular data source, see [Reloading data for a single data source](#) on page 228 instead.



You can also choose to reload data for a single data table instead, from the [Data canvas](#) on page 178. There, you can select whether to reload **Linked data** only (sources with 'Always new data' or 'New data when possible'), or **All data** in the data table (including 'Stored data' and 'Embedded in analysis').

### Prerequisites

You must have some data loaded in the analysis. To reload data from all possible data sources you must have a Spotfire Analyst or a Spotfire Business Author role. If you only have a Spotfire Consumer role, or if the analysis is in Viewing mode, you cannot reload data from data sources using stored data or top-level embedded data, only linked data.



Because reloading can affect both stored and top-level embedded data you must consider this if you are an author and it is important that the analysis should contain specific data. In some cases, you might want to break the link to the original source to prevent others from reloading data unintentionally.

### Procedure

- On the menu bar, select **Data** and one of the reload options below:
  - **Reload linked data** – All linked data with the data loading settings 'Always new data' or 'New data when possible' will be updated. However, linked data sources that have data loading set to [Stored data](#) will not be reloaded using this procedure.
  - **Reload all data** – All data that is possible to reload will be reloaded, including linked data with data loading set to 'Stored data' and embedded data from sources that are possible to reload (provided that the source is available). If you have a data source with data from the clipboard, or, if you have frozen some columns in an embedded data table, the data table cannot be reloaded at all. See below for more information about embedded data. This option is only available to authors.



Reloading all data tables may take a long time if one or more data tables are very big. If data is saved linked, you can select exactly which [part of a data table to reload](#) instead.



Reload of in-database data tables will only reload the data, not the schema. See [Data Connection Settings](#) in the *Spotfire Analyst User Guide* if you need a full schema refresh.



If an analysis has been configured to use scheduled updates, you will not be able to reload linked data. Scheduled updates are normally used to save time for end users by downloading the latest data before anyone needs it. Such analyses often contain large amounts of data which should not be reloaded randomly. Read more about scheduled updates in the Spotfire Server [Installation and Administration Help](#).



If you want to edit transformations before the reload, you can do this from the data canvas using the installed client. To modify import settings for a source (for example, for an Excel file), you can replace the data source with itself.

## Embedded data

It is possible to embed a data table in the analysis so that the analysis file is self-contained with data and possible to use offline. This can be done from the data canvas, by clicking **Settings** and choosing **Embedded in analysis**.

Embedded data captures a snapshot of the final data table in the analysis file. When data is saved embedded, you cannot reload different parts of the data table separately, as in the linked case. However, the **Reload all data** option can be used to reload data from the original data source, provided that this source is available. This means that the embedded data might not necessarily be fixed and unchanged during the lifetime of an analysis. If a reload is done, then the latest data from the original source is fetched into the analysis. Hence, the analysis will contain a snapshot of the data from the latest time of reload and save. Simply opening an analysis with embedded data will not cause a reload of the data.



If you have a data source in your data table that you currently cannot reload, you might be able to recreate a link by replacing the data source with another source (for example, not clipboard data).

See also [Load methods](#) on page 58 and [Linked, stored, and embedded data](#) on page 70.

## Reloading data for a single data source

When you are working with data from Salesforce, Google Analytics or saved data files in the library, you can keep the data linked to the source, so that the data is reloaded from the source each time you open the analysis. However, you may also want to reload the linked data during a session, without having to reopen the analysis. For example, if you know that data in the source is refreshed every five minutes and you want to view the absolutely latest data, then you can reload the data for a particular data source as described in this topic.

If you want to reload more than one data source in your analysis, see [Reloading data](#) on page 227 instead.


### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.



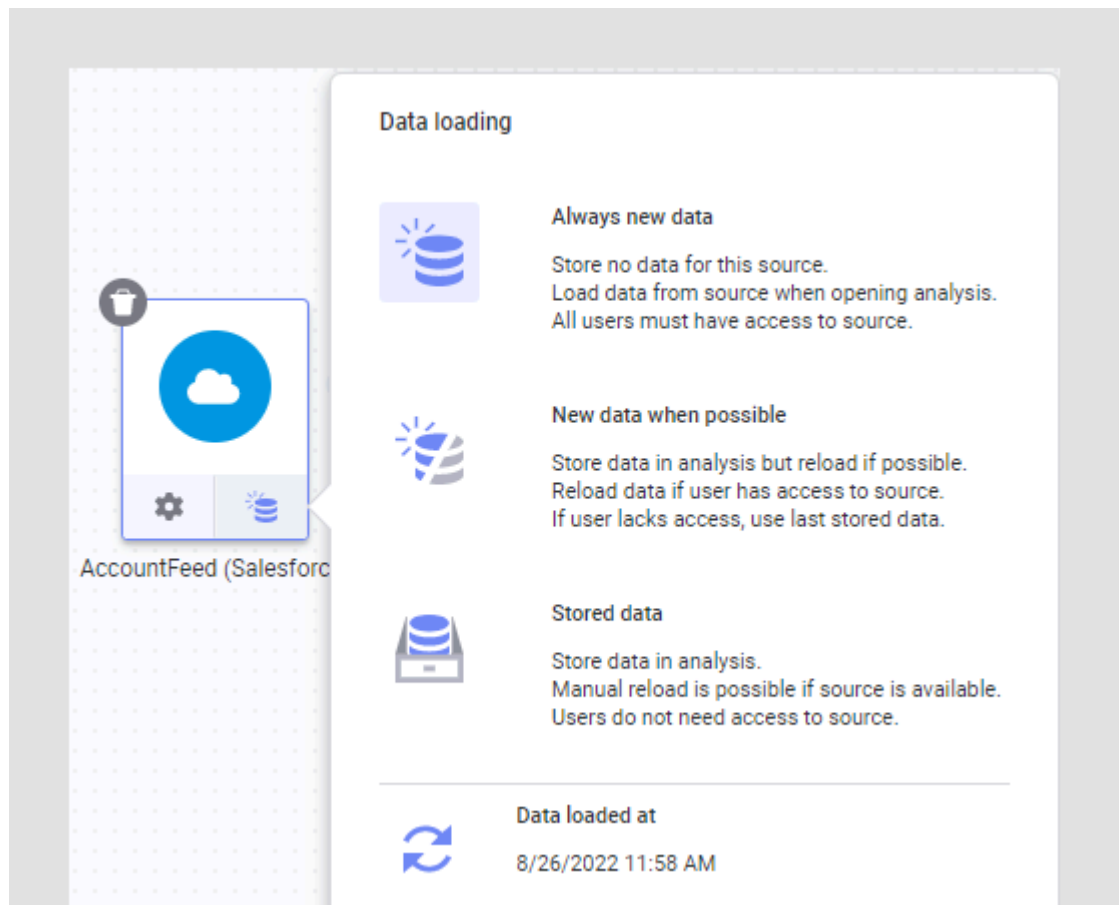
You need access to the Spotfire Business Author license to perform these steps.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.

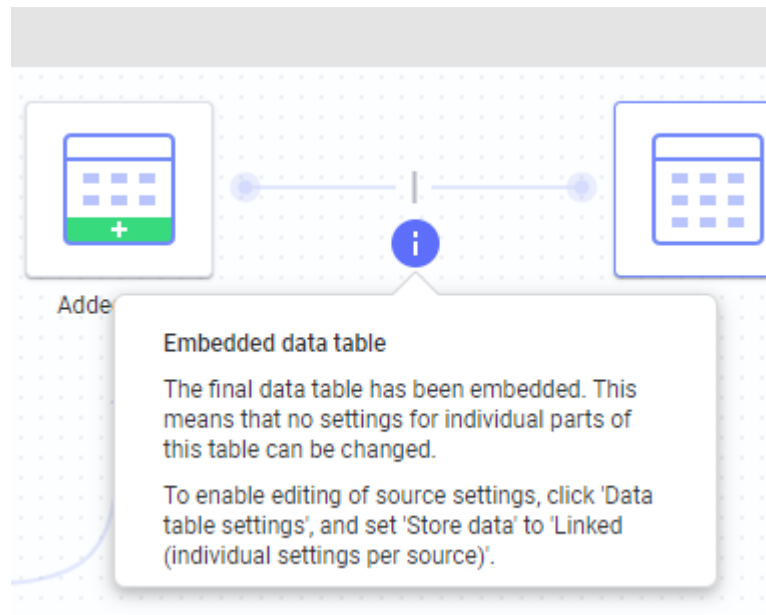


3. Locate the node that represents the data source you want to reload and click on the button at the lower part of the node representing the data loading settings.




If your data source lacks this type of button, then it means that the data source cannot be reloaded. For example, if the final data table has been saved as embedded (under **Settings** in the [Data canvas](#) on page 178), then you will not see any button at all on the data source. Read more about

embedded data under [Linked, stored, and embedded data](#) on page 70 and [Reloading data](#) on page 227.



Also, calculated data sources based on data functions or other tools in Spotfire Analyst, or data that can only use stored data (for example, data pasted from the clipboard in Spotfire Analyst or data from local files that have been uploaded to the web client) cannot be reloaded.

4. In the **Data loading** popover, click the **Reload data** button, .

## Replacing a data table

You can replace a data table in your analysis with data files that are saved locally on your computer, data that has been previously saved to the library, or, if you have access, you can use data from available data connections.

It is possible to reuse the visualizations, calculations and setup from a previously created analysis with new data, as long as the new data is reasonably similar to the old data. If the new data differs much from the previous data you might need to recreate new visualizations from scratch.

When you replace the data table, only the operations done on the final data table (such as added calculated columns, data types changed from the column view, etc.) will be kept, whereas transformations, added rows or columns, and other intermediate operations will disappear.



If you want to keep transformations, added data and other intermediate operations (all steps between the first added data source and the final data table in the source view), you should consider [replacing the data source](#) instead of replacing the entire data table.



Local data files to be used can be in different formats: Microsoft Excel Workbooks (.xlsx, .xls), Microsoft Excel Binary Workbooks (.xlsb), Microsoft Excel Macro-Enabled Workbooks (.xlsm), comma separated values files (.csv), text files (.txt), logfiles (.log), Spotfire Text Data Format file (.stdf) and Spotfire Binary Data Format file (.sbdf).

Data can be saved to the library using Spotfire Analyst, in the form of Spotfire Binary Data Format files (.sbdf), data connections, or information links.

### Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.

## Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.
3. On the data canvas toolbar, click **Replace** .
4. In the **Files and data** flyout, select the new source data. Click **OK** when you are satisfied.  
Depending on the selected source type, you might be able to change the load method before clicking OK (choose between **Import** (in-memory analysis) or **External** (in-db analysis)).



When a data connection is added to the analysis, all views in the connection are available within the analysis until the connection or a view is removed using **Data > Data Connection Properties** (installed client only), regardless of whether the view is added as a data table or not. Therefore, you should select a connection under **Data connections in current analysis** (in the **Files and data** flyout, under **Recommended**) to add more views from an already used connection, rather than adding another connection to the same source.

The data source is replaced with new data.

If some used columns cannot be replaced automatically by columns in the new data table, you are sometimes presented with a dialog where you can match columns from the current data table with columns from the new data table (installed client only). Manual updates might still be needed.

5. If necessary, update any visualizations, calculations, or hierarchies that were broken when the data was replaced.



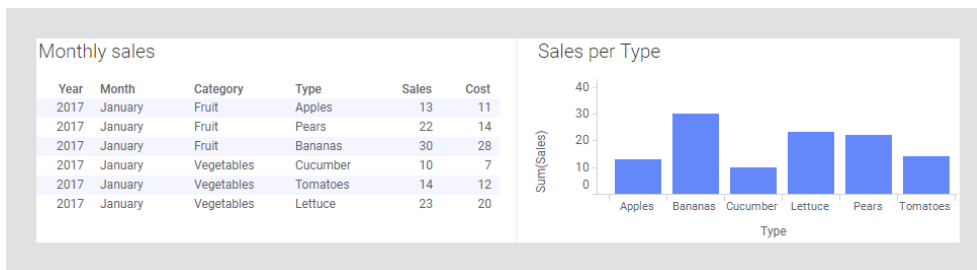
If you replace a data table used in expressions containing preprocessor syntax, all such expressions must be manually updated to continue working. See [Properties in expressions](#) on page 907 and [Troubleshooting property expressions](#) on page 912 for more information.

## Result

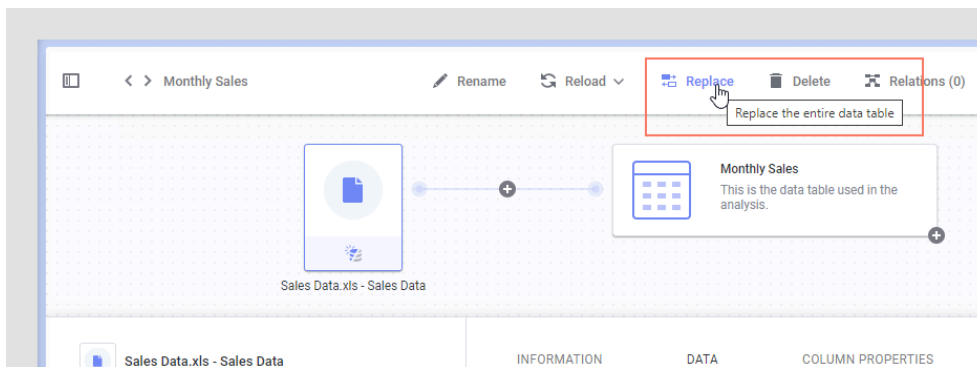
The analysis uses data from the new data source.

### Example: Replacing sales data for a new month

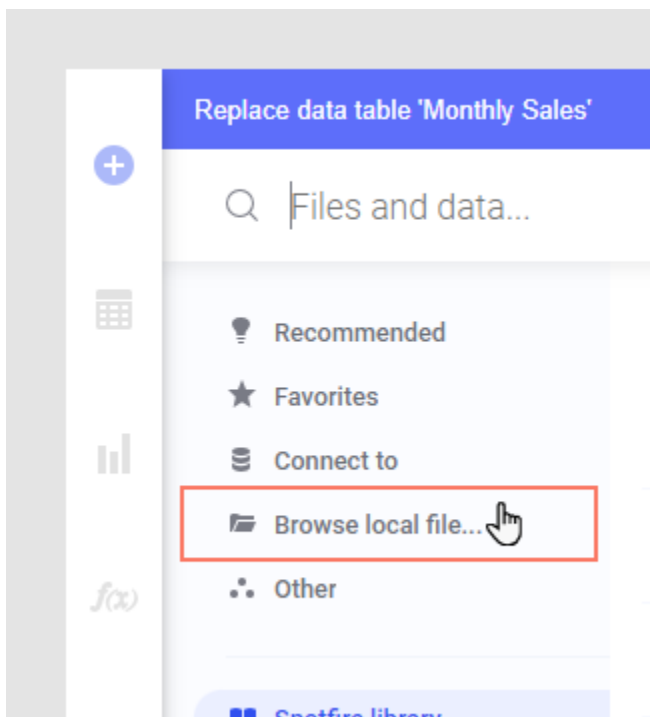
Replacing is useful when creating an analysis for, say, sales figures for a certain month. You can create a full analysis using the data from January, set up visualizations, calculations, etc., and save the file.



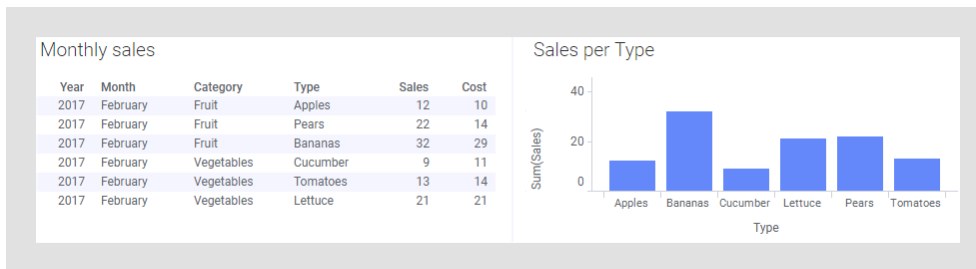
When the sales figures for February are available, you can open the same file again, and, in the **Data canvas**, replace the data from January with the data from February, and the visualizations will be updated.



Select from where to fetch the new data.



If you use a local file, such as a Microsoft Excel Workbook, you might need to specify which worksheet to use before the data is replaced.



This of course requires that the data table for February is structured in the same way as for January, using the same column names and format. If the column names should differ, you must modify the visualizations to use the new column names after the data has been replaced.

## Replacing a data source

A data table can be based on data from one or more data sources. You can replace an individual data source in the data canvas if the data table is linked and not top-level embedded. This is often preferred over replacing the entire data table.

It is possible to reuse the visualizations, calculations and setup from a previously created analysis with new data, as long as the new data is reasonably similar to the old data.

By replacing a data source, rather than [replacing the entire data table](#), you can keep the data table intact. That means that all transformations, calculations, added rows or columns, and operations that were used to build the final data table are kept as is. Use these steps to switch from a test database to a production database, or when your data remains exactly the same but is moved from one type of data source to another.

Note that if subsequent transformations or calculations become invalid due to changes in the structure of the new source (e.g., different column names), then the resulting data table might become empty, that is, without any data or columns. It is always a good idea to review the [data canvas](#) after a replace operation to look for errors or warnings.

When you replace a data source (e.g., from a local file data source to a source file in the library), the data loading behavior will be switched to the default setting for the selected type of source. In most cases, the data loading behavior can be changed by clicking on the bottom part of the source node in the source view, should the new default not be what you wanted.


### Prerequisites

You must have some data imported into the analysis, and the analysis must be in **Editing** mode.

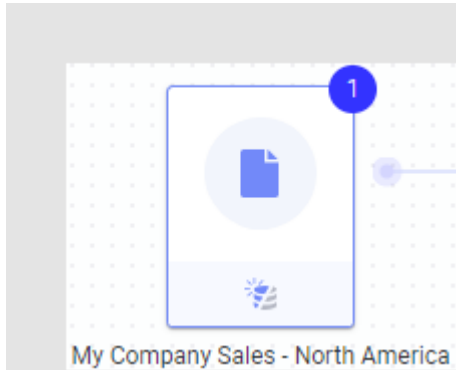


To replace an external data source, or to change the type of data from imported (in-memory) data to external data (in-db), you must [replace the entire data table](#) instead.


### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected.  
This step is only applicable if you have two or more data tables in the analysis.

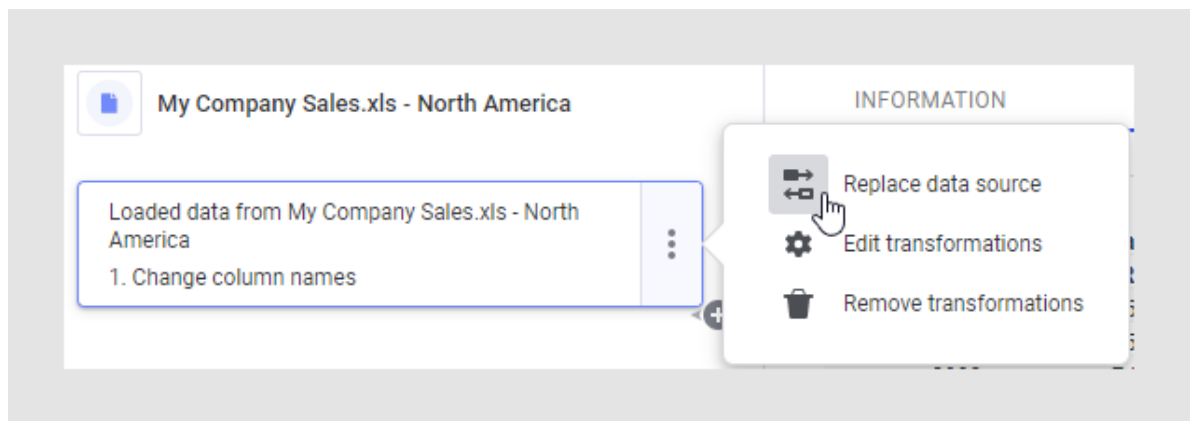
3. In the data canvas, click on the node representing the data source to replace.



Details about the selected node are shown in the bottom left part of the data canvas.

4. Locate the 'Loaded data' step and click **Replace data source** .

If transformations are available, you must first click the menu on the Loaded data step, before you can choose the replace option.



(Editing transformations is available in the installed client only.)

5. Choose the new data and close all dialogs and flyouts.



When a data connection is added to the analysis, all views in the connection are available within the analysis until the connection or a view is removed using **Data > Data Connection Properties** (installed client only), regardless of whether the view is added as a data table or not. Therefore, you should select a connection under **Data connections in current analysis** (in the **Files and data** flyout, under **Recommended**) to add more views from an already used connection, rather than adding another connection to the same source.

The data source is replaced with new data. If you select a new data source that does not support transformations directly on the data source, the transformations will still be kept, but they will be moved to a separate group in the data source history.

If some used columns cannot be replaced automatically by columns in the new data table, you are sometimes presented with a dialog where you can match columns from the current data table with columns from the new data table (installed client only). Manual updates might still be needed.

6. If necessary, update any transformations, visualizations, calculations, or hierarchies that were broken when the data was replaced.
7. If desired, modify the data loading behavior for the new source in the [Data canvas](#) on page 178.

## Result

The analysis uses data from the new data source.

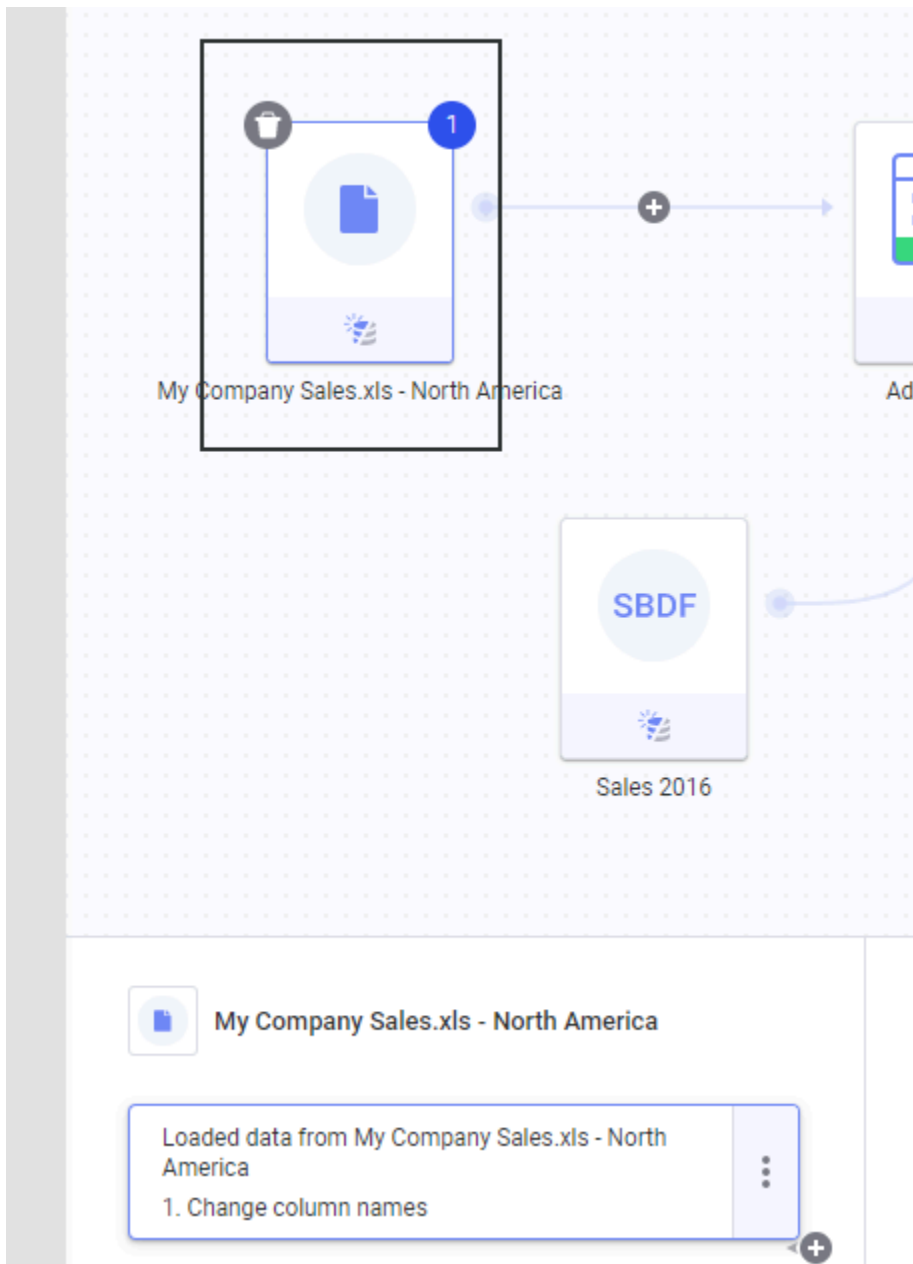
### Example: Switching from a local data source to a Salesforce connection


In this example, someone has configured an analysis with good visualizations using local data from Excel, and you want to replace the data with fresh data using your Salesforce account.

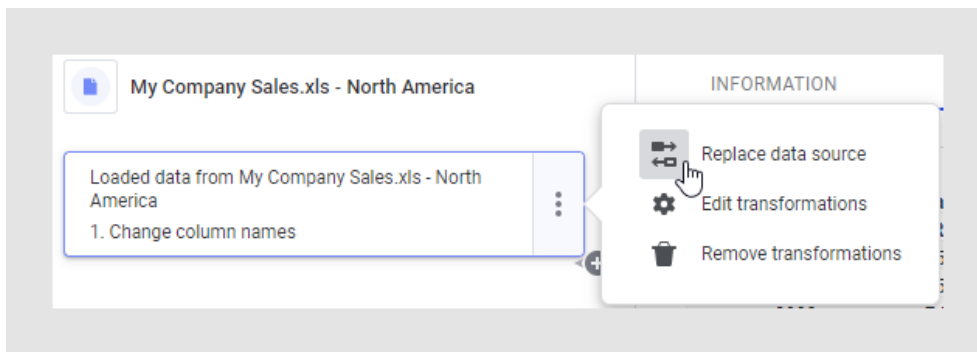
Make sure the analysis is in **Editing** mode and, on the authoring bar, click **Data**

canvas .

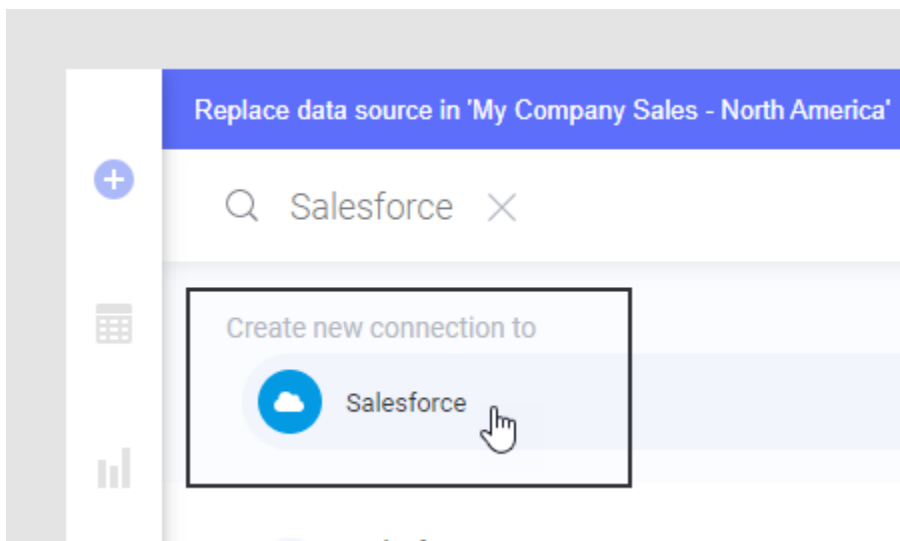
Click on the interesting node.



In the lower part of the data source, click **Replace data source**  next to the 'Loaded data' step. If a transformation is available you must first click the menu to see the option.

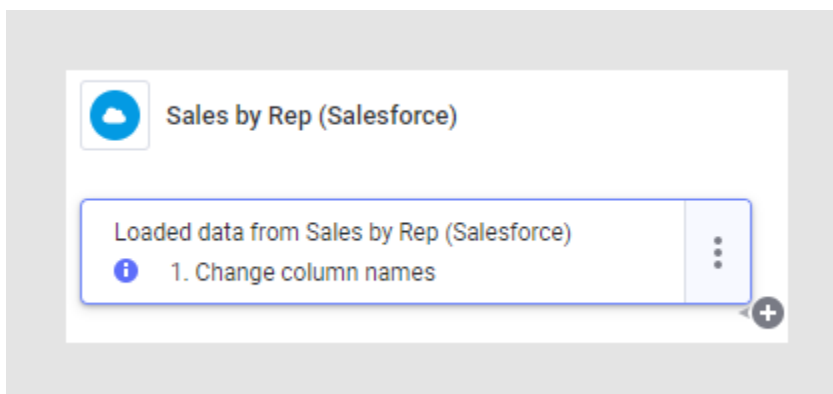


In the **Files and data** flyout, select the new data by creating a new connection or using a predefined connection to the new data source (Salesforce), logging in, and browsing to the interesting data. See [Creating a new analysis from Salesforce data](#) for details.



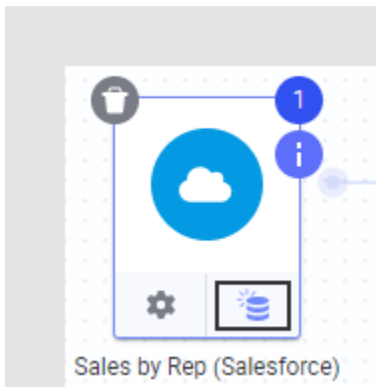
Click **OK** to load the new data.

In this example, the old transformation used on the previous data source was kept when the data source was replaced.

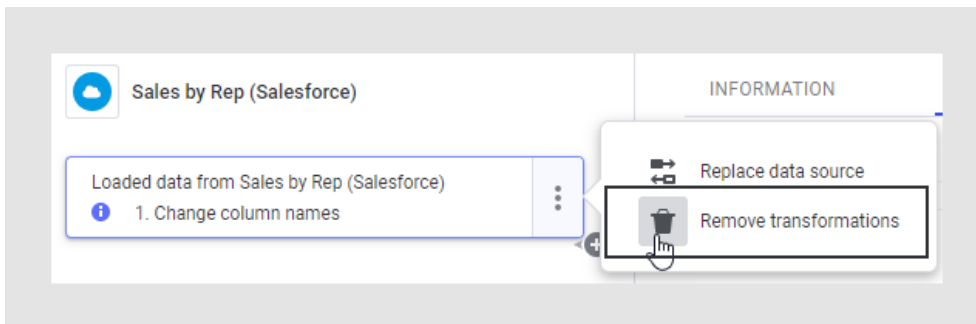


Also note that the data loading behavior has changed from the previously specified 'New data when possible' to 'Always new data', which is the default data loading behavior for a data connection:





As it turns out, the transformation is no longer needed, as indicated by the tooltip on the information icon, and it can easily be removed:



(The **Remove** icon will not be available if you cannot reload the source, that is, you must log in to the new source to be able to remove the transformation.)

The data has now been replaced and the analysis is ready to be used.

## Replacing empty values in a column

When you have empty values in your data, you can easily replace them in the Spotfire clients.



### Prerequisites

Your data with one or several missing values must be loaded in the analysis and the analysis must be in **Editing** mode.

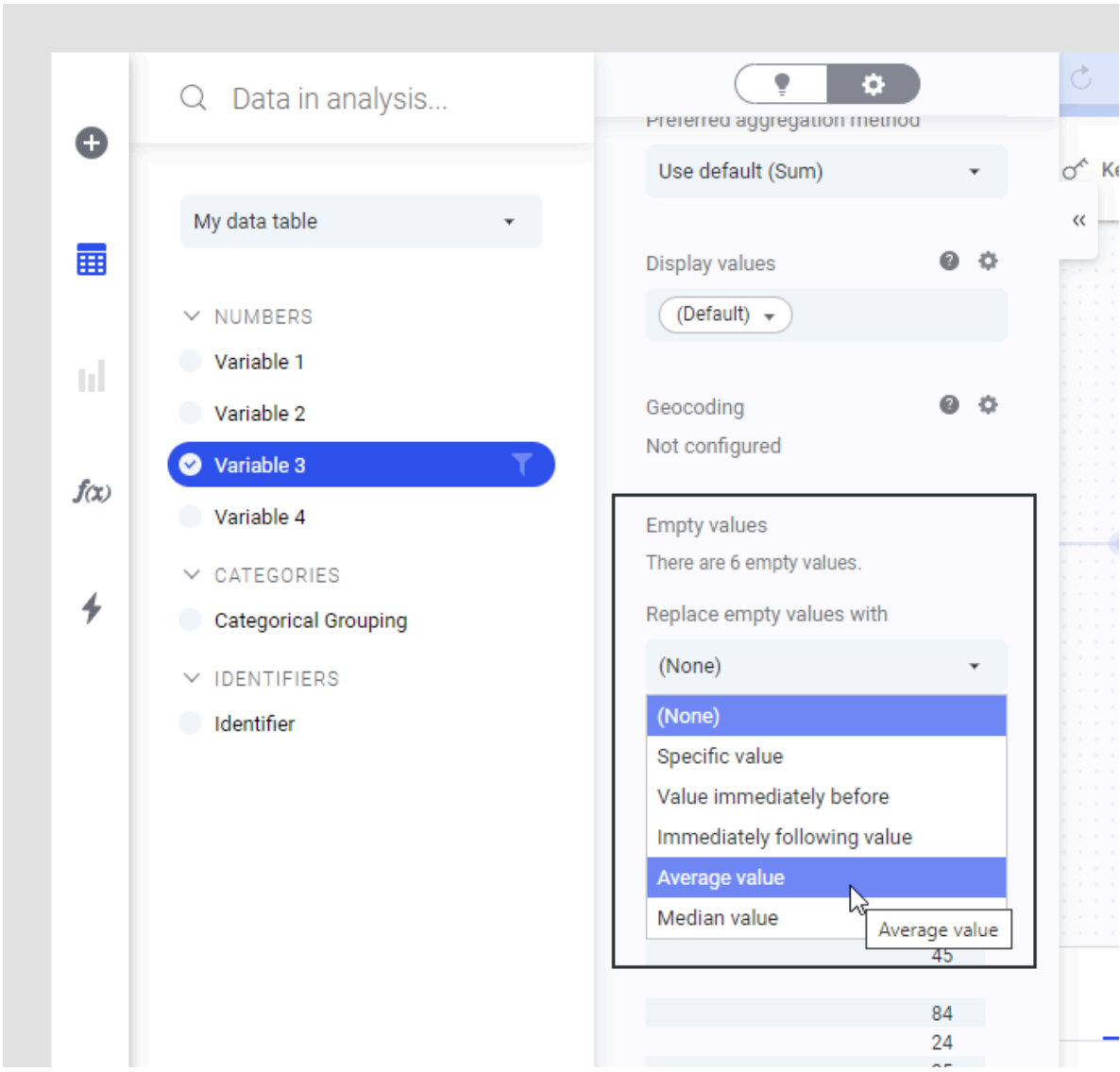


You can also handle empty values by [replacing the values](#) directly in a table visualization.



### Procedure

1. On the [authoring bar](#), click **Data in analysis**  and click on the column of interest.
2. In the expanded flyout, make sure that **Details on selected column**  is selected.
3. In the expanded flyout, scroll down and locate the **Empty values** section.

- Under **Replace empty values with**, select one of the options.



Option	Description
(None)	No empty values are replaced.
Specific value	Allows you to type the value to insert instead of the missing values. <div> <div>7</div> <div>8</div> <div></div> <div>3</div> </div> <span>→</span> <div> <div>7</div> <div>8</div> <div>0</div> <div>3</div> </div>
Value immediately before	Replaces empty values with the first valid value that comes immediately before the missing value in the original data. <div> <div>7</div> <div>8</div> <div></div> <div>3</div> </div> <span>→</span> <div> <div>7</div> <div>8</div> <div>8</div> <div>3</div> </div>

Option	Description																																				
	<p>This can be particularly interesting if the original data is an Excel file with a structure like the one below, where one value should apply to all of the following empty rows:</p> <div><table><tr><th>Department</th><th>Name</th><th>Salary</th></tr><tr><td>Sales</td><td>John Doe</td><td>3000</td></tr><tr><td></td><td>Jane Doe</td><td>4500</td></tr><tr><td></td><td>James Smith</td><td>3500</td></tr><tr><td>Engineering</td><td>Sue Johnsson</td><td>3100</td></tr><tr><td></td><td>Robert Baker</td><td>4000</td></tr></table><p>→</p><table><tr><th>Department</th><th>Name</th><th>Salary</th></tr><tr><td>Sales</td><td>John Doe</td><td>3000</td></tr><tr><td>Sales</td><td>Jane Doe</td><td>4500</td></tr><tr><td>Sales</td><td>James Smith</td><td>3500</td></tr><tr><td>Engineering</td><td>Sue Johnsson</td><td>3100</td></tr><tr><td>Engineering</td><td>Robert Baker</td><td>4000</td></tr></table></div>	Department	Name	Salary	Sales	John Doe	3000		Jane Doe	4500		James Smith	3500	Engineering	Sue Johnsson	3100		Robert Baker	4000	Department	Name	Salary	Sales	John Doe	3000	Sales	Jane Doe	4500	Sales	James Smith	3500	Engineering	Sue Johnsson	3100	Engineering	Robert Baker	4000
Department	Name	Salary																																			
Sales	John Doe	3000																																			
	Jane Doe	4500																																			
	James Smith	3500																																			
Engineering	Sue Johnsson	3100																																			
	Robert Baker	4000																																			
Department	Name	Salary																																			
Sales	John Doe	3000																																			
Sales	Jane Doe	4500																																			
Sales	James Smith	3500																																			
Engineering	Sue Johnsson	3100																																			
Engineering	Robert Baker	4000																																			
Immediately following value	<p>Replaces empty values with the first valid value that comes immediately after the missing value in the original data.</p> <div><table><tr><td>7</td><td>7</td></tr><tr><td>8</td><td>8</td></tr><tr><td></td><td>3</td></tr><tr><td>3</td><td>3</td></tr></table><p>→</p><table><tr><td>7</td><td>7</td></tr><tr><td>8</td><td>8</td></tr><tr><td>3</td><td>3</td></tr><tr><td>3</td><td>3</td></tr></table></div>	7	7	8	8		3	3	3	7	7	8	8	3	3	3	3																				
7	7																																				
8	8																																				
	3																																				
3	3																																				
7	7																																				
8	8																																				
3	3																																				
3	3																																				
Average value	<p>Replaces empty values with the average value of the other values in the column. For integer values, the average is rounded towards zero when needed.</p> <div><table><tr><td>7</td><td>7</td></tr><tr><td>8</td><td>8</td></tr><tr><td></td><td>6</td></tr><tr><td>3</td><td>3</td></tr></table><p>→</p><table><tr><td>7</td><td>7</td></tr><tr><td>8</td><td>8</td></tr><tr><td>6</td><td>6</td></tr><tr><td>3</td><td>3</td></tr></table></div> <div><p>If you want to avoid rounding of the average value you can <a href="#">change the data type</a> of the column to Real before replacing the empty value .</p></div>	7	7	8	8		6	3	3	7	7	8	8	6	6	3	3																				
7	7																																				
8	8																																				
	6																																				
3	3																																				
7	7																																				
8	8																																				
6	6																																				
3	3																																				
Median value	<p>Replaces empty values with the median value of the other values in the column. For integer values, a calculated median is rounded towards zero when needed.</p> <div><table><tr><td>7</td><td>7</td></tr><tr><td>8</td><td>8</td></tr><tr><td></td><td>7</td></tr><tr><td>3</td><td>3</td></tr></table><p>→</p><table><tr><td>7</td><td>7</td></tr><tr><td>8</td><td>8</td></tr><tr><td>7</td><td>7</td></tr><tr><td>3</td><td>3</td></tr></table></div> <div><p>If you want to avoid rounding of the median value you can <a href="#">change the data type</a> of the column to Real before replacing the empty value .</p></div>	7	7	8	8		7	3	3	7	7	8	8	7	7	3	3																				
7	7																																				
8	8																																				
	7																																				
3	3																																				
7	7																																				
8	8																																				
7	7																																				
3	3																																				

Refer to [Calculated Columns](#) for more information about the syntax of the mentioned [Expressions](#) and [Functions](#).

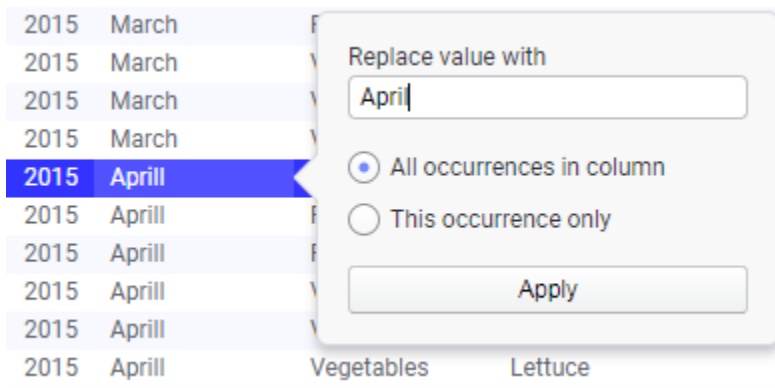
## Replacing a value

If you find that a value in your data table is misspelled, or, if one value has been entered on a different form than the other values in the column, you can replace that value directly from a table visualization, from the Details-on-Demand, or from the column overview in the expanded Data in analysis flyout.

### Prerequisites

Imported data (not in-database data) must be loaded in the analysis and the analysis must be in **Editing** mode.

In a [table visualization](#), in the [Details-on-Demand](#), or in the [expanded data in analysis flyout](#), double-click on the value you want to replace and type a new value.



These steps are used to replace a few values in a column only, not to [change the formatting of a column](#) or to [replace an entire data table](#). If you want to combine several different categories into one, you might want to use [Group from marked categories](#) instead.

The change is applied as a transformation and does not affect the source data. For details about the different options, see the corresponding section.

### Details

The underlying mechanism behind the Replace Value shortcut is a transformation. If you want to include some type of replaced value in a script, or when using the API to create custom tools, you can add similar transformations by using an expression.

For example, if [PK] is a column which uniquely identifies rows (that is, a primary key column) and you want to replace the specific value 'a' with 'b' on the row where [PK] is 34, you can write the following expression:

```
case
  when [PK] = 34 and [Col] = 'a' then 'b'
  else [Col]
end
```

To replace all occurrences of 'a' with 'b' in [Col], write:

```
case
  when [Col] = 'a' then 'b'
  else [Col]
end
```

### Replacing all occurrences of a value in a column

You can use **Replace value** with the **All occurrences in column**-option to replace all instances of a value directly from a table visualization, from the Details-on-Demand, or from the column overview in the expanded Data in analysis flyout.

The procedure for replacing a single occurrence of a value in a column is quite similar, but that requires that a key column has been specified for the data table. See [Replacing a specific value in a column](#) for more information.



These steps are used to replace a few values in a column only, not to [change the formatting of a column](#) or to [replace an entire data table](#). If you want to combine several different categories into one, you might want to use [Group from marked categories](#) instead.

### Prerequisites

Imported data (not in-database data) must be loaded in the analysis and the analysis must be in **Editing** mode.

## Procedure

1. In a [table visualization](#), in the [Details-on-Demand](#), or in the [expanded data in analysis flyout](#), double-click on the value you want to replace.

You can also right-click and select **Replace value** from the pop-up menu.

2. Type a new value in the popover.
3. Make sure that **All occurrences in column** is selected.
4. Click **Apply**.

## Result

All instances of the value are replaced in the column and any visualizations using the data are updated.

The replacement is added as a transformation on top of your data. The original data in your data source is not affected by this change. This means that if the underlying data is reloaded, the transformation will be reapplied after the reload and any additional rows matching the rule will also be replaced in the analysis. If the underlying value is changed in the source data, so that the transformation is no longer applicable, you will see an indication about this in the [data canvas](#).



You can remove a previously added 'replace value'-operation from your data table. See [Removing operations in the data canvas](#) on page 194 for more information.



You can use one of the procedures under [Exporting data](#) on page 951 to save the cleansed data table for reuse in other analyses.

## Replacing a specific value in a column

You can use **Replace value** with the **This occurrence only**-option to replace a specific value directly from a table visualization, from the Details-on-Demand, or from the column overview in the expanded Data in analysis flyout.

### Prerequisites

Imported data (not in-database data) must be loaded in the analysis and the analysis must be in **Editing** mode.

If you want to replace just a single occurrence of a value in a column, you must be able to identify the row, even if the data is reloaded. This is done by specifying key columns for the data table. Key columns can be specified by clicking **Key columns** in the data canvas toolbar. See also [Specifying key columns for a data table](#) on page 199.

See also [Replacing all occurrences of a value](#) in a column.



These steps are used to replace a few values in a column only, not to [change the formatting of a column](#) or to [replace an entire data table](#). If you want to combine several different categories into one, you might want to use [Group from marked categories](#) instead.

## Procedure

1. In a [table visualization](#), in the [Details-on-Demand](#), or in the [expanded data in analysis flyout](#), double-click on the value you want to replace.

You can also right-click and select **Replace value** from the pop-up menu.

2. Type a new value in the popover.
3. Make sure that **This occurrence only** is selected.
4. Click **Apply**.

## Result

The value is replaced and any visualizations using the data are updated.

The replaced value is added as a transformation on top of your data. The original data in your data source is not affected by this change. This means that if the underlying data is reloaded, the transformation will be reapplied after the reload. If a previous key column identification is broken by the reload, so more than one value is affected by the transformation, you will be informed about this in the [data canvas](#). Also, if the underlying value is changed in the source data, so that the transformation is no longer applicable, you will see an indication about this in the data canvas.



You can remove a previously added 'replace value'-operation from your data table. See [Removing operations in the data canvas](#) on page 194 for more information.



You can use one of the procedures under [Exporting data](#) on page 951 to save the cleansed data table for reuse in other analyses.

## Replace value – more information

An analysis can contain a few different column types where 'replace value'-transformations cannot be used. This topic explains why values in these columns cannot be replaced using the **Replace value** option, and what you can do instead, if a value seems wrong.



Many of the column types described here can only be added or edited in the installed client, not in the web client. See the [Spotfire Analyst](#) help for more information.

For more information about the **Replace value** option, see [Replacing a value](#) and its related topics.

## Calculated columns

Calculated columns are created using an expression which often includes the values in other columns. If a specific value in the calculated column looks wrong, it may be the value in one of the original columns that should be replaced instead.

You can edit or review the expression to detect the original columns included in an expression by clicking on the calculated column in the **Data in analysis** flyout, and, in the [column view of the expanded Data in analysis flyout](#), under **Expression**, click **Edit**. See [Editing a calculated column](#) for more information.

The limitation also applies to calculated columns that are created using a tool, such as binned columns, or columns created using a [split](#) operation in the column view.

## Columns with column conversions

If you have [changed the data type](#) in the [column view of the expanded Data in analysis flyout](#), you must first reset the data type to the original data type before you can replace the value. However, you can reapply the data type change after the value has been replaced.

Similarly, it will not be possible to replace values in a column where [empty values](#) have been replaced from the column view. Apply any type of column conversion after the replacement has been done, or add the desired conversion as a transformation on the column rather than using the quick column conversion methods in the column view (transformations are only available in the installed client).

## External data

It is not possible to replace values when the data from a [data connection](#) is external. In-database data is generally expected to come directly from the data source, so if there is an error in that data it should probably be fixed by the database administrator in the data source rather than in the analysis.

However, it is sometimes possible to import the data from the data connection instead of keeping it external. Authoring of data connections is, in most cases, handled using the installed client.

### Result columns

Columns that are an output from another operation do not support value replacement. Like the case with calculated columns, any issues with result columns may actually be an issue with an original input column used in the calculation.

Depending on the type of result column, you may be able to edit the input to the calculation and get an updated result by refreshing the visualization containing the result values.

However, for most types of result columns you need to run the tools again, with different settings, to produce a different result. For more information, see [What are Data Functions?](#), [How to Perform a K-means Clustering](#), [How to Perform a Line Similarity Comparison](#), [What is the Hierarchical Clustering Tool?](#) and [What is Predictive Modeling?](#) in the Spotfire Analyst help.

### Tags columns

[Tags](#) columns are manually created columns, where selected rows have been assigned a specific value, a 'tag'. If you want to change the tag, open the **Tags** panel (**View > Tags**) and right-click on the tag, select **Edit tag** and change the tag name there.

### Binary columns

Values in columns containing binary data, such as images or geometries, cannot be replaced.

### Column titles

When information about a single row is shown in the Details-on-Demand, the column names from the data table are displayed in a column. You cannot change column names using **Replace value**. Instead, open the **Data in analysis** flyout, select the data table of interest, right-click on the column and select **Rename**. You can also edit the column name from the details view in the expanded Data in analysis flyout.

## Document properties, column properties and data table properties

Properties are metadata about different parts of an analysis file. Some of the properties are created automatically by Spotfire or by other tools that produce data, but when you are working with an analysis file in a Spotfire client, you also have the option to add custom properties to the analysis. Properties can be used inside expressions in different parts of an analysis, such as in calculated columns or custom expressions, and also be used with property controls in text areas (added using the installed client). Data functions can use properties both as inputs and outputs.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

There are three different types of properties where you can add a custom property: document properties, data table properties and column properties. They have slightly different purposes but can be used in similar ways.

- **Document properties** apply to the whole document and are often used as a variable value. They can be used in calculations, for selections of certain values, or directly in visualizations as target values or similar.
- **Column properties** contain metadata about a specific column or hierarchy, and they can define things like the expression or the description of a column, min and max values within a column, the type of column, and so on.

- **Data table properties** contain metadata about data tables (or data sources within a data table), such as, what type of data table it is, or how it has been created.

Data table and column properties can have [default values](#). If a default value has been defined, then the default value for the data table or column will be used unless a specific value has been defined. See [Editing data table or column property values](#) on page 247 for more information.



If you change the default value, it will be updated and used as the default value for both new data tables and already created data tables.

## Document properties

Document properties are the most used type of custom properties. They apply to the entire document and can contain any type of value that you want to be able to reuse in the document, like a global variable.

For example, it could be a max, min or target value for some number that you want to visualize as a line in a visualization, or use to filter or color values by. It can also be used as a variable in calculations, like an exchange rate, or similar. If you are trading with foreign countries, and the figures you want to analyze depend on the current exchange rate, then you can create a property that represents the exchange rate, and use the property instead of a fixed exchange rate in all conversion expressions. When the exchange rate changes, you adjust the value of the property to the current rate, and recalculations are performed at all places where the property is involved.

When properties are used in expressions, you can select to either use the current value of the property or to add it as text. This makes it possible to also use properties to represent columns in the analysis. See [Properties in expressions](#) on page 907 for more information.

New document properties and their values can be added manually. See [Adding or editing a document property](#) on page 245 for more information. They can also be specified as the output from a data function calculation. Document property values can be updated by an end user's selections in a property control, if the analysis author has specified this. Read more about the latter in [Using document, data table or column properties in an analysis](#) on page 250 and the following topics.

## Column properties

Column properties contain metadata about a specific column or hierarchy, and can be used to classify columns of a certain type, or add specific information about how the column is built or should be used. They can define things like the origin of a column, the description of a column, min and max values within a column, whether or not a column is a geocoding column, and so on. Column properties are sometimes automatically generated from the data source, but they can also be manually created, using any client.

Custom column properties are added to all columns in the analysis, including those in other data tables, but you do not need to add a value for columns where the property is not applicable. The default value of a string property can be empty. Custom column properties can be used to create groups of columns for use in property controls.

The column properties can be used when you create reference lines in visualizations, or, they can be a part of an expression. See [Using document, data table or column properties in an analysis](#) on page 250 for some examples where properties can be used.

There are some column properties available that can point out that a column can be used for [geocoding](#) map data. Even though geocoding information can be specified directly from the **Data in analysis** flyout, you can also make changes to geocoding properties from the data canvas.

In the data canvas, there is an overview of how your columns were created and you can see the values of all column properties during each step (for each node in the graphical source tree). You can edit the value for a specific column by clicking on the final data table node and entering edit mode. See [Editing data table or column property values](#) on page 247 for more information.



## Data table properties

Data table properties contain metadata about a specific data table and are not as commonly used as the other types of properties. They can be used to classify different types of data tables.

For example, data table properties are used to specify that a data table is a [geocoding](#) data table, which can help placing data correctly on a map. Read more in [Specifying new geocoding tables](#) on page 172.

When data tables have been added using information links, the keywords and description defined in Information Designer will also be displayed as data table properties.

The [data canvas](#) gives you an overview of how your data tables were created and shows the values of all data table properties for each node in the graphical source tree. You can edit the value for a specific data table when clicking on the final data table node and entering edit mode. See [Editing data table or column property values](#) on page 247 for more information.

## Adding or editing a document property

Document properties can either be added manually, updated using a property control, or specified as the output from a data function calculation.

If you want to add a new document property and its value manually, you can use the Document properties panel. In the Document properties panel, you can also edit or delete properties, or get an overview of where different properties are used.

See also [Document properties](#) for general information.

### Prerequisites

You must be an author with the Advanced Document Properties license feature. You can only use the Document properties panel when in Editing mode.

To add or edit text areas with property controls, you must use the installed client.

### Procedure

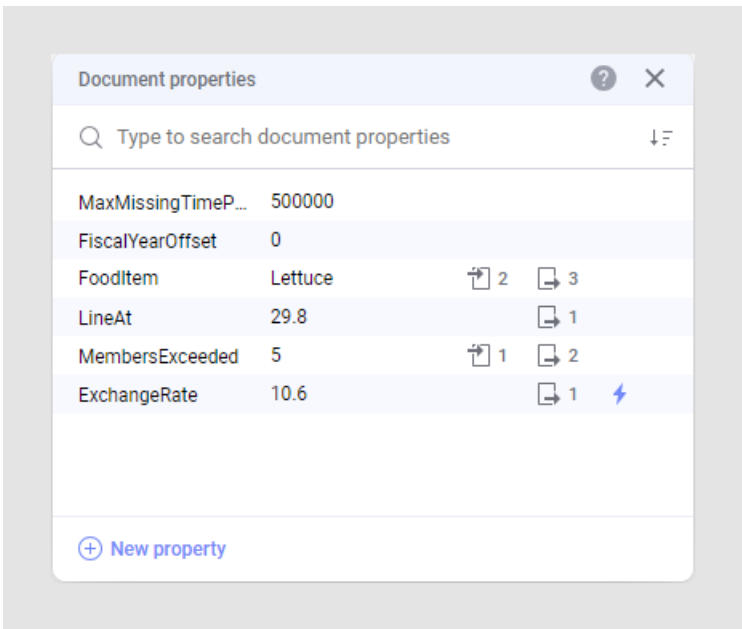
1. On the menu bar, select **View > Document properties**.
2. In the Document properties panel, click **New property**.
3. In the **New document property** dialog, enter a name and specify a data type. Optionally you can add a description and enter a value.
4. Click **OK**.



You can also open the Document properties panel at a later time and select **Edit** or **Edit value** from the context menu to open the **Edit document property** dialog or to only change the value.

### Result

The new document property is added to the list, and can be used as a global variable. The variable value can be used in calculations, expressions, as an input value for a data function or action, or directly in visualizations as a target value or similar. See [Using document, data table or column properties in an analysis](#) on page 250 and the following topics for some examples.



## Adding a new data table or column property

You can add your own data table or column properties from the **Data tables** overview page, reached from the data canvas sidebar.

### Prerequisites

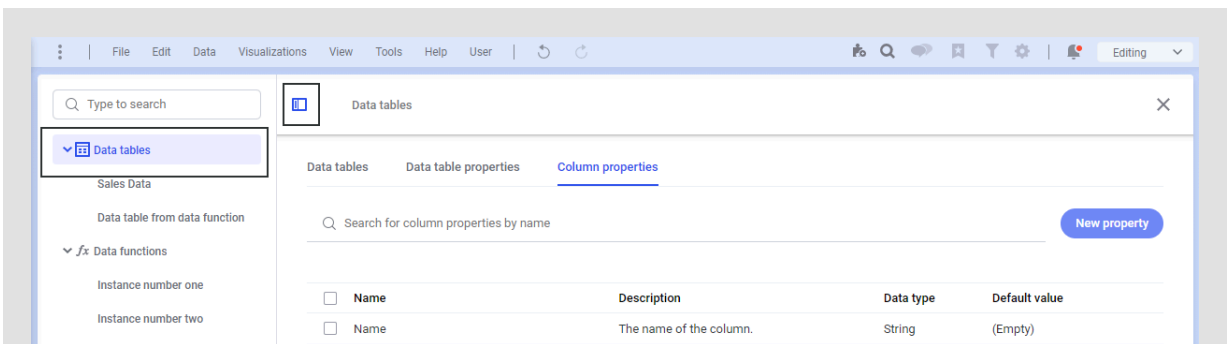
You are an analyst or business author with the **Manage Data Tables** license feature. If an analysis is open, it must be in **Editing** mode.





New properties can also be created in some places where you can use them. For example, in the installed client, you can add a new property by right-clicking in the **Available properties for column** list in the expression dialogs, selecting **New**, and specifying a property type.



See [Using Properties in the Analysis](#) in the *Spotfire Analyst User Guide* for some examples of where you can use properties.



### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. If the data canvas sidebar is collapsed, click **Show the data canvas sidebar** .
3. In the data canvas sidebar, click **Data tables**.

4. Choose which type of custom property to add by going to the corresponding tab, **Data table properties** or **Column properties**.
5. Click **New property**.
6. In the new property dialog, enter a **Property name**.
7. Select a **Data type** for the new property.



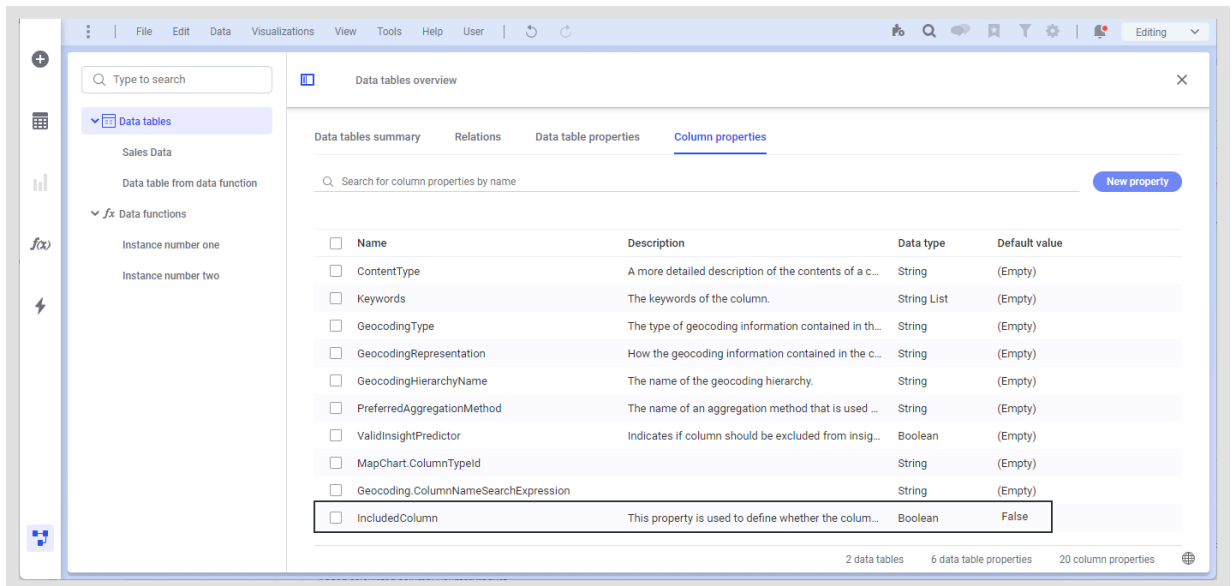
Some types of properties, for example, properties used for [geocoding](#), must be of a specific data type so make sure you select the correct type.

8. Add a **Description** so others can understand the meaning or use of the property.
9. If the property needs a default value, make sure **Use default value** is selected and enter the default value to use.
10. Click **OK**.

## Result

The new property is added to the list of available properties.

By default, the new properties are added to the end of the properties list, but you can click on a header to sort the list by that column, or use the search field to find your newly added property.



To specify a property value for a column or a data table, see [Editing data table or column property values](#) on page 247.

## Editing data table or column property values

You can edit the value of a property for a specific column or data table in the lower right part data canvas. Editing is possible for properties that you have created yourself, as well as for some properties that are automatically added to the analysis (such as the name or the description of a column or data table).

### Prerequisites

You are an analyst or business author with the **Manage Data Tables** license feature. An analysis must be open and it must be in **Editing** mode.

Data table and column properties can have default values. If a default value has been defined, then the default value for the data table or column will be used unless a specific value has been defined, as described below.




You can change the default value (as well as the description of the property) for a data table or column property from the **Data tables** overview in the data canvas sidebar. See [Changing a property default value](#) on page 249 for more information. If you change the default value, it will change both the value for new data tables or columns and already created data tables or columns that are currently using the old default value. If other settings are wrong, you must delete the property and create a new one instead.



In the installed client, you can often edit the value of a property in the places where you can use them. For example, you can edit properties by right-clicking in the **Available properties for column list** in the expression dialogs.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected in the data canvas and choose which type of custom property to edit by going to the corresponding tab in the lower right part of the data canvas, **Data table properties** or **Column properties**.
3. Click on the final node in the graphical structure.



You can only edit data table or column properties on the final node (representing the data table used in the analysis). If you look at data table or column properties for earlier nodes in the data source tree, all values are read-only.

4. Click the switch next to **Edit properties** to enter the edit mode.
5. If you are editing column properties, you must select which column to update the property for using the **Edit property values for the column** drop-down list.



You can search for a column name, or a part of a column name, by typing directly in the column selector.

6. Double-click the property to update in the list, or click to select the property and then click **Edit** above the table.



Use the search field to search for the property name.

7. In the Edit property value dialog, make sure the **Specify a value** check box is selected, and type the value you want to use in the field below. For string values, you can use an empty string.
8. Click **OK**.


### Result

The value of the column or data table property is updated for the selected column or data table.

## Changing the link template for hyperlinks in a table visualization

Using the installed client, it is possible to show the text in a table visualization as clickable hyperlinks by setting the renderer for a column to **Link renderer**. A template in column properties provides a default suggestion for the link renderer. The default template is `http://{ $ } /`, where { \$ } represents the value from the table cell.

You can change the link template for a specific column using the following steps:

1. On the [authoring bar](#), click **Data canvas** .
2. Make sure the data table of interest is selected in the data canvas.
3. Click to select the final data table node (to the right in the graphical structure) and then click on the **Column properties** tab in the lower right part of the data canvas.
4. Locate the column of interest by typing the column name in the list, or by browsing the list.
5. Locate the **LinkTemplate** property (for example, using search) and click to select it.
6. Click **Edit**.
7. Make sure **Specify a value** is selected and type a new template for the format of the links.

For example, `https://www.{ $ }.com`.

8. Click **OK**.



If you want to change link renderer settings for the whole application instead of just for a single column, this can be specified using the installed client under **Tools > Options > Application > Renderer Settings > Add > Add Default Renderer**.

## Changing a property default value

You can change the default value for most data table or column properties from the **Data tables** overview in the data canvas. List-valued properties created in the text area and properties of the data type Binary cannot be edited.

If you change the default value, it will change both the value for new data tables or columns and already created data tables or columns that are currently using the old default value.



Most property types allow you to change the default value and the description. If other settings are wrong, you must delete the property and create a new one instead.


To change the property value for a specific column or data table, see [Editing data table or column property values](#) on page 247 instead.

### Prerequisites

You are an analyst or business author with the **Manage Data Tables** license feature. If an analysis is open, it must be in **Editing** mode.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .

2. If the data canvas sidebar is collapsed, click **Show the data canvas sidebar** .
3. In the data canvas sidebar, click **Data tables**.
4. Choose which type of default value to update by going to the corresponding tab, **Data table properties** or **Column properties**.
5. Locate the property to edit in the list and click to select the row for the property.



Use the search field if you have many properties.

6. Click **Edit** above the table.
7. Update the **Description**.
8. Make sure that **Use default value** is selected and enter the new default value.
9. Click **OK**.

## Using document, data table or column properties in an analysis

A property, especially a document property, can be used as a variable to drive configurations of visualizations and calculations and make them easy to modify by others. Adding a property control that can change the property value to a text area (installed client only) further simplifies the update process. The property controls can be used to give web client consumer users a possibility to change the axes of visualizations or modify the analysis in other ways.

This section collects different ways that you can use properties to enhance or simplify a Spotfire analysis.

### Prerequisites

Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

For example, a document property can be used to specify a column name. This property can then be used to define what is shown on one or more visualization axes, either directly or as a part of a custom expression. By using a property instead of simply placing the column name directly on the axes, you only need to change a single value (the property value) to change all axes where the property is used.

Property expressions can also be used to define a line or a curve. If a property used in expressions is updated, the property will be updated in all currently used locations.

The process to follow when setting up property controls is usually as follows:

1. Decide which values in the analysis people should be able to control from the text area.
2. Create a property, which can assume these variations in values, and its property control.
3. Hook the property to the place where user control is wanted, for example, an axis selector or an expression.

The best way to explain these steps is through examples. They also give you a hint of what is possible to do.

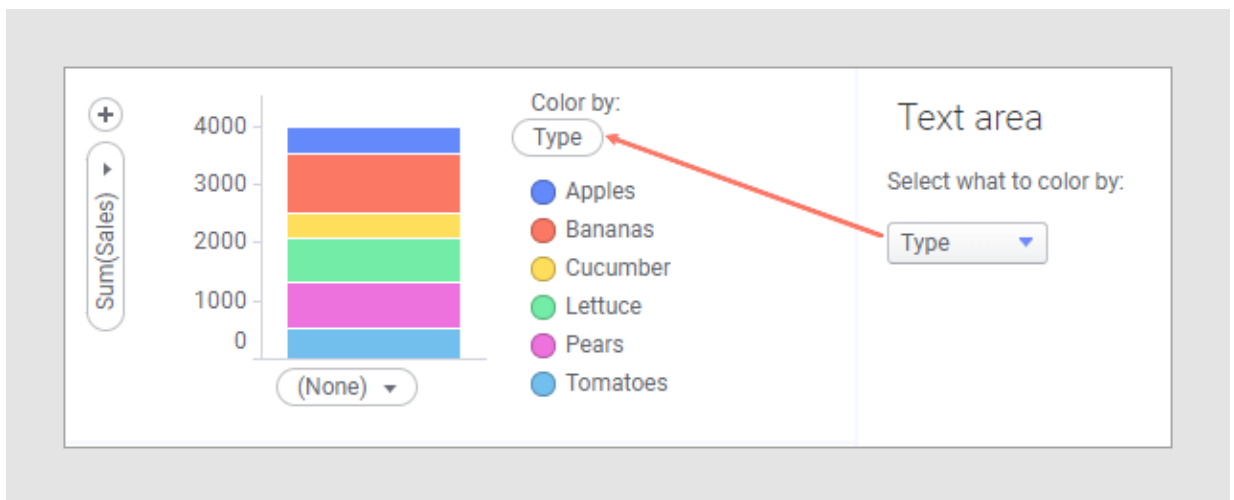
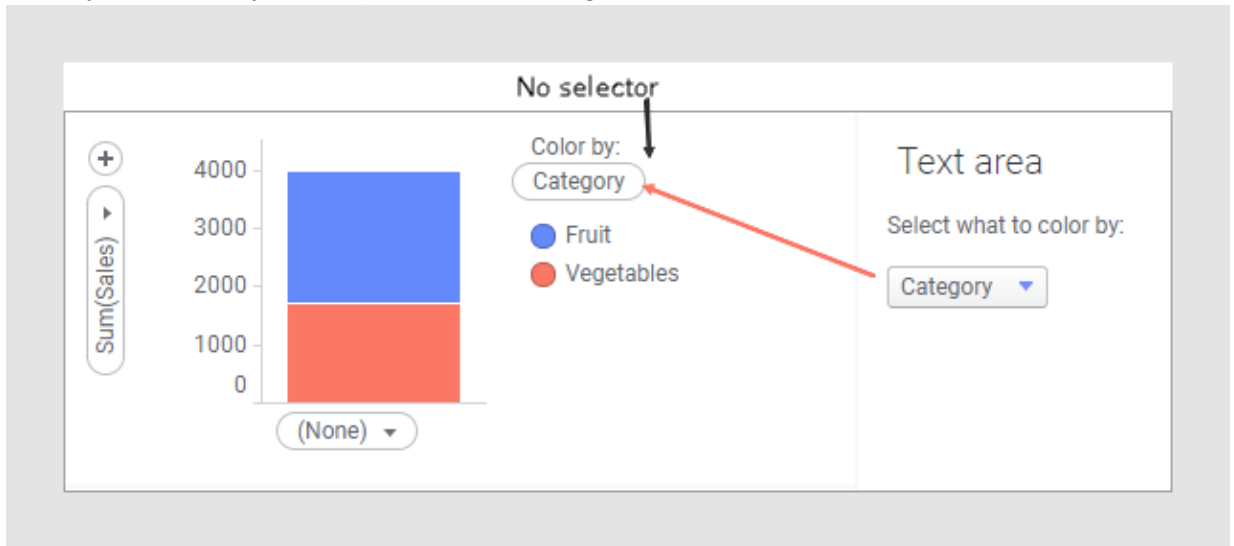
### Handling column selections using a property

This is an example of how you can use a document property to control the column selection of an axis from a text area.

#### Prerequisites

The functionality described here must be authored using the installed client.

In the bar charts below, the column to color the bar chart by is selected from the text area rather than being configured on the **Color by** selector in the visualization. The text area contains a property control, here in form of a drop-down list, where a dashboard user can choose which column to color the bar chart by. The ordinary **Color by** selector is no longer available in the bar chart.



Follow the steps to create this type of analysis.

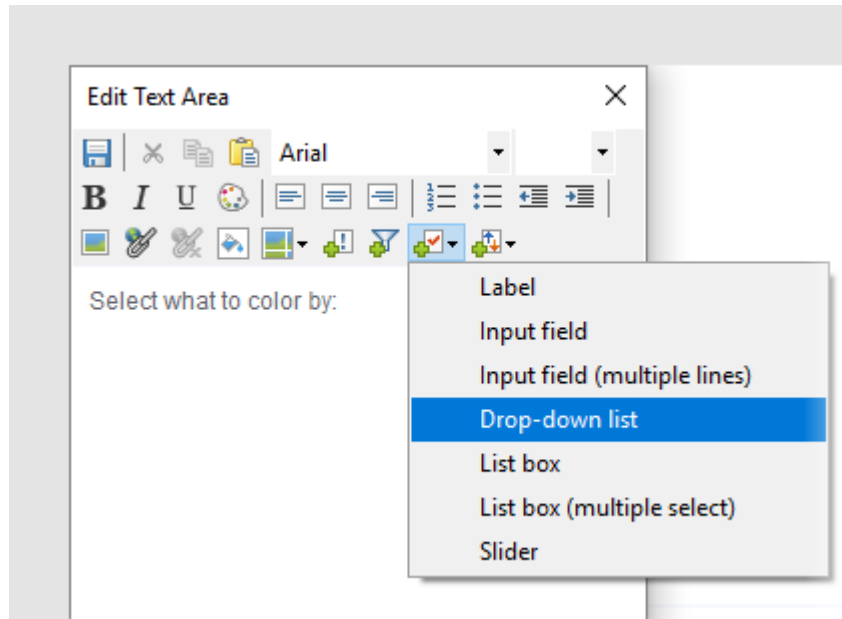
### Prerequisites

Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

### Procedure

1. Create or edit a [text area](#).

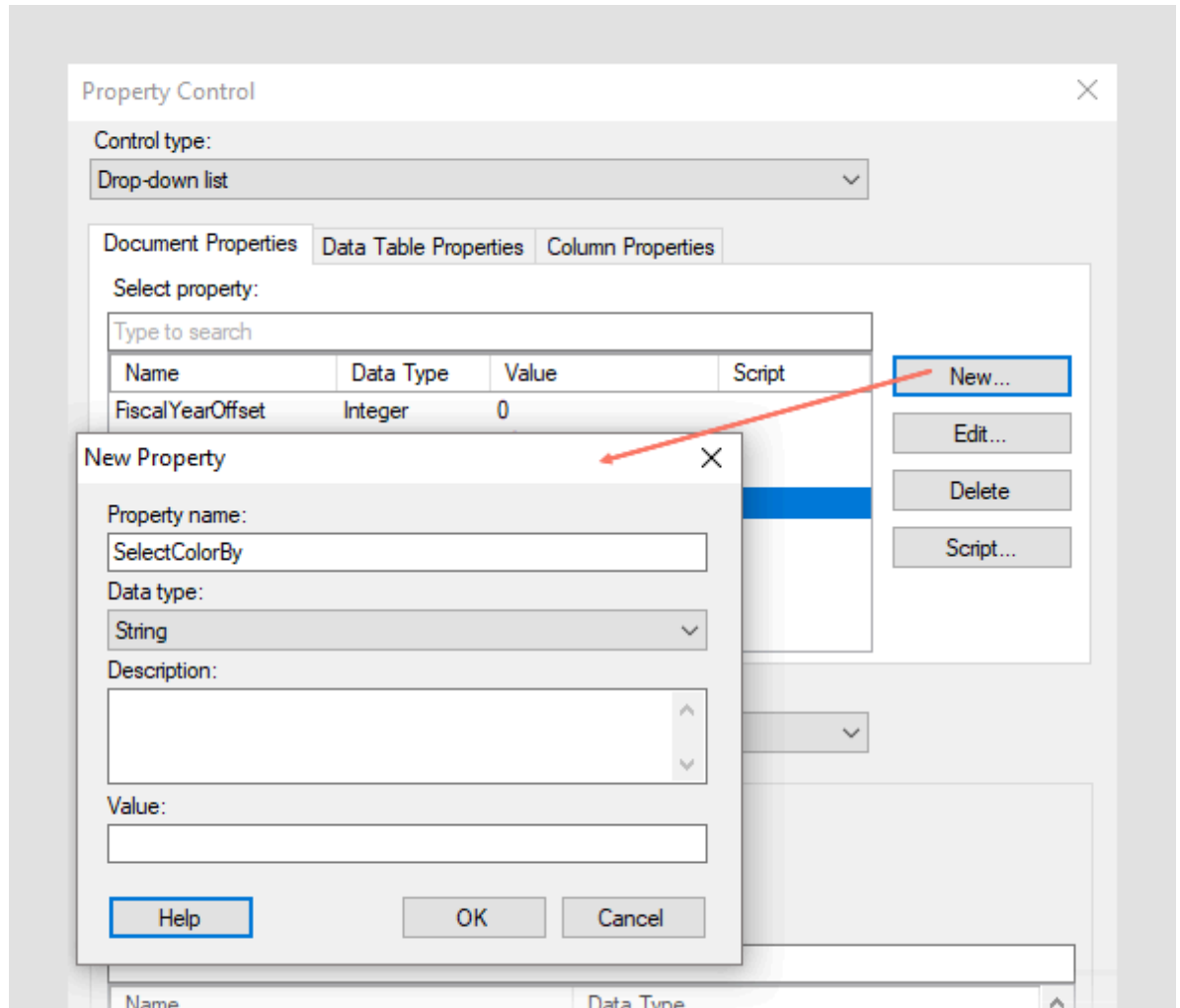
2. In edit mode, [add the property control](#) to the text area. In this example, we add a drop-down list.



3. In the Property Control dialog, the items that should be available for selection in the drop-down list are defined. However, the actual property to be controlled does not yet exist. To create it, click **New**



to open the New Property dialog, where you give the property a name and specify its data type. In this example we call it 'SelectColorBy'.



4. When the property exists and has been selected, define the values the property should be able to have. In this example, the value of the property is either 'Category' or 'Type', as you want to color the bar chart by either of these columns. The settings are made in the lower part of the Property

Control dialog. These property values are set through **Column selection**, and an 'or' expression is used to define the two columns to include.

Select Columns By: String

Delete

Script...

Set property value through:

Column selection

Settings

Selectable columns (limit through expression):

Name	Data Type
Category	String
Type	String

☐ Include (None) alternative

Select Columns...

For more information on search expressions click Help.

Help

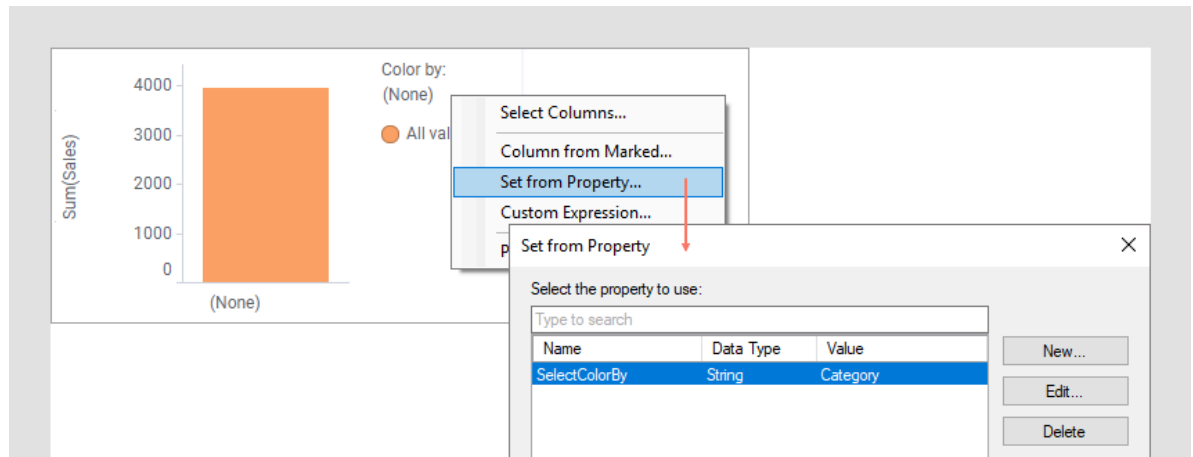
OK

Cancel



You can use any search expression to determine which columns to show in a property control. For example, use `datatype:integer` to show integer columns only, use `datatype:real` to show real columns only, use `"A or B"` to show columns beginning with the letters A or B only, and so on. You can also search for a particular column property. The **Select Columns** dialog is a shortcut to creating column properties and selecting columns based on these properties. For example, use `included.column:true` if you have added a custom column property called `included.column` with the default value `false` and the value `true` for those columns you want to be available. See also [Searching in Spotfire clients](#) on page 956 for more information about search expressions.

- The last step is to connect the property to the place that should react on the selected property value, in this case the **Color by** selector in the bar chart. Right-click the selector, choose **Set from Property**, and select the new property.



### Using multiple document properties to determine the columns to show on an axis

You can add more than one property to an axis expression, if you want the axis to show multiple values or a hierarchy.

#### Procedure

- Right-click the axis selector for the axis of interest and select **Custom Expression** from the pop-up menu.
- Insert the property of interest into the expression using the text syntax. For example, `${MyProperty}`.

For more information about the different syntaxes available for properties, see [Properties in expressions](#) on page 907.

See [Adding or editing a document property](#) on page 245 for information about how to add a new property. Note that the data type of this type of property must be *String*.

- Add as many properties as you like to the expression.  
For a continuous axis with multiple columns based on different properties, separate each property with a comma:

```
${MyProperty}, ${MyProperty2}, ${MyProperty3}
```

For a categorical axis, specify how the categories should be handled using [NEST or CROSS](#):

```
<[${MyProperty}] NEST [${MyProperty2}] NEST [${MyProperty3}]]>
```

If the string value consists of several words (e.g., "My Value"), then you must use `$esc()` or put brackets around the property expression to return a column: `[${MyProperty}]`.

- If desired, you can edit the expression for more complex calculations. When you are ready, click **OK**.

#### Result

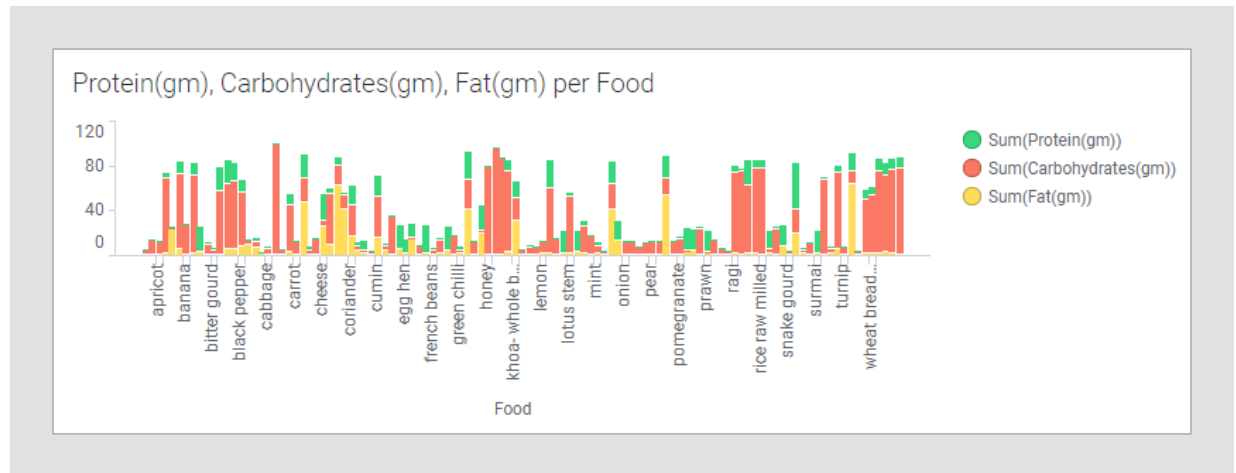
The visualization axis uses the default value of the properties as different columns. You can add property controls in a text area, as described in the previous example, for easy update of the property values.

## Limiting data using a property expression

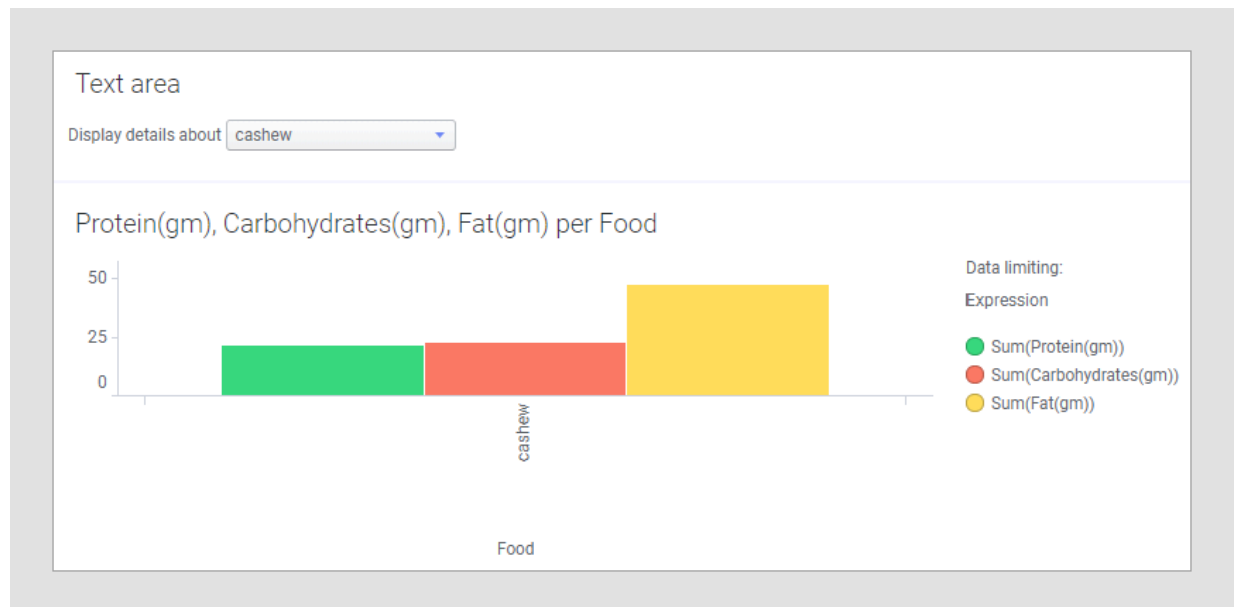
In this example, a property expression is used to pick out details for a certain category from a bar chart that is otherwise quite packed with information. The stacked bars show the amounts of protein, carbohydrates, and fat in different items of food.

### Prerequisites

The functionality described here must be authored using the installed client.

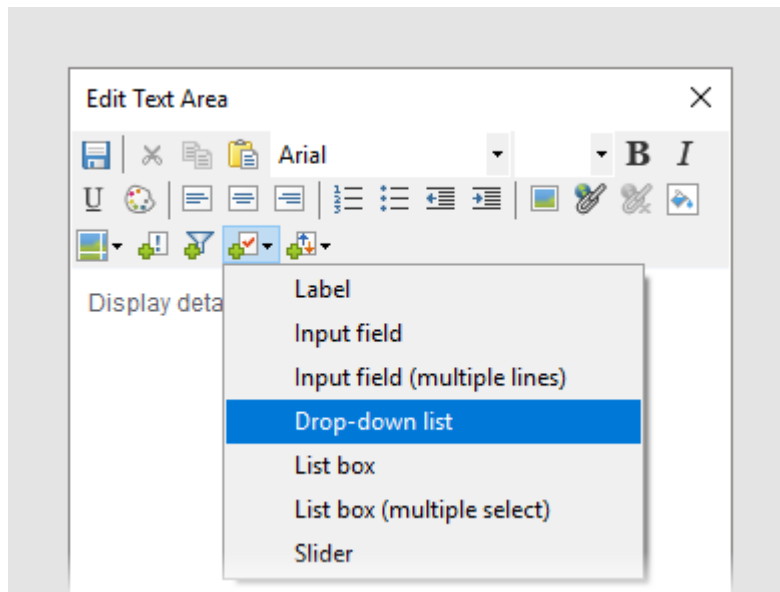


Assume you want to look closer at one item at a time, and you want to specify which item to look at from a text area, as shown below.

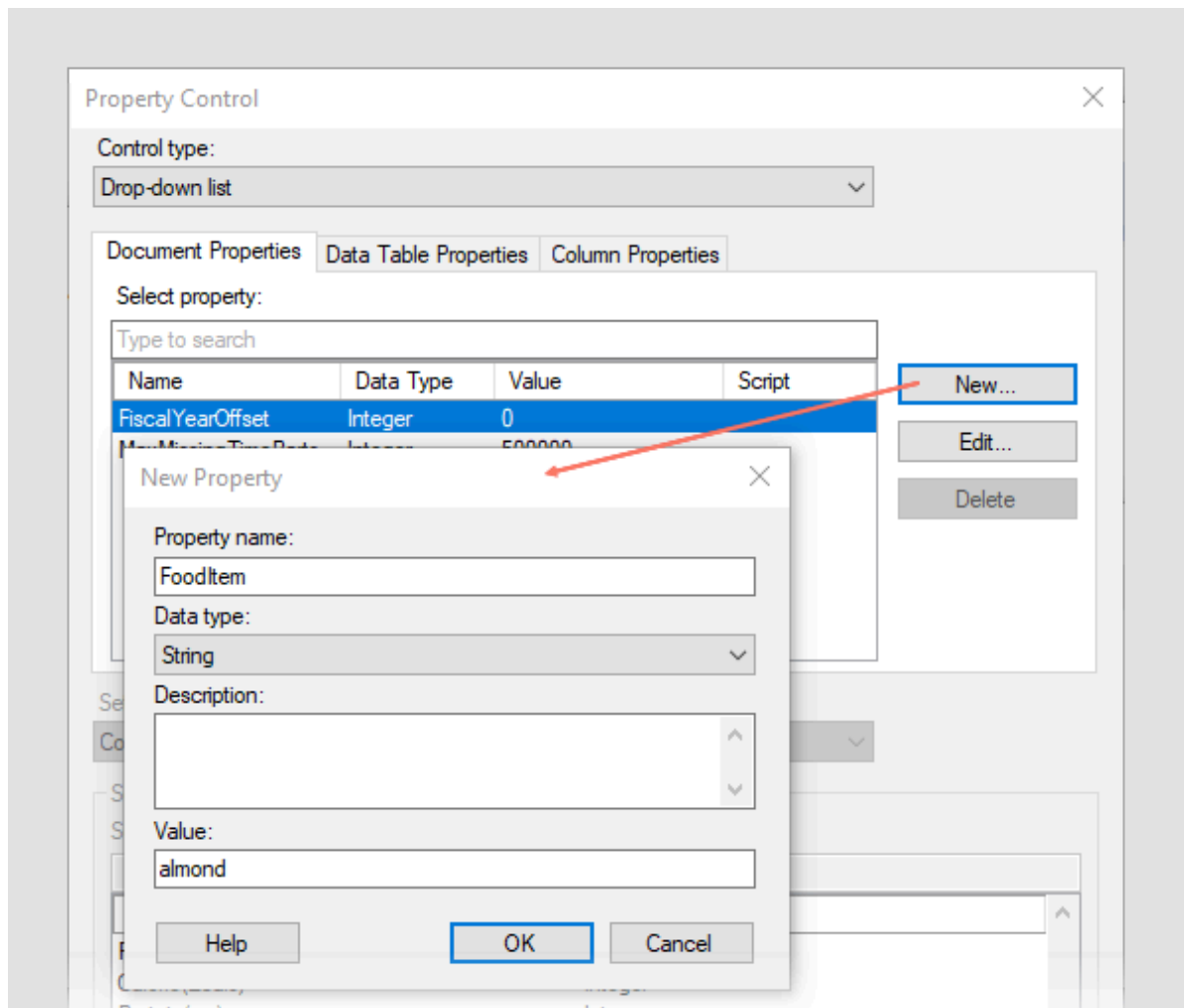


## Procedure

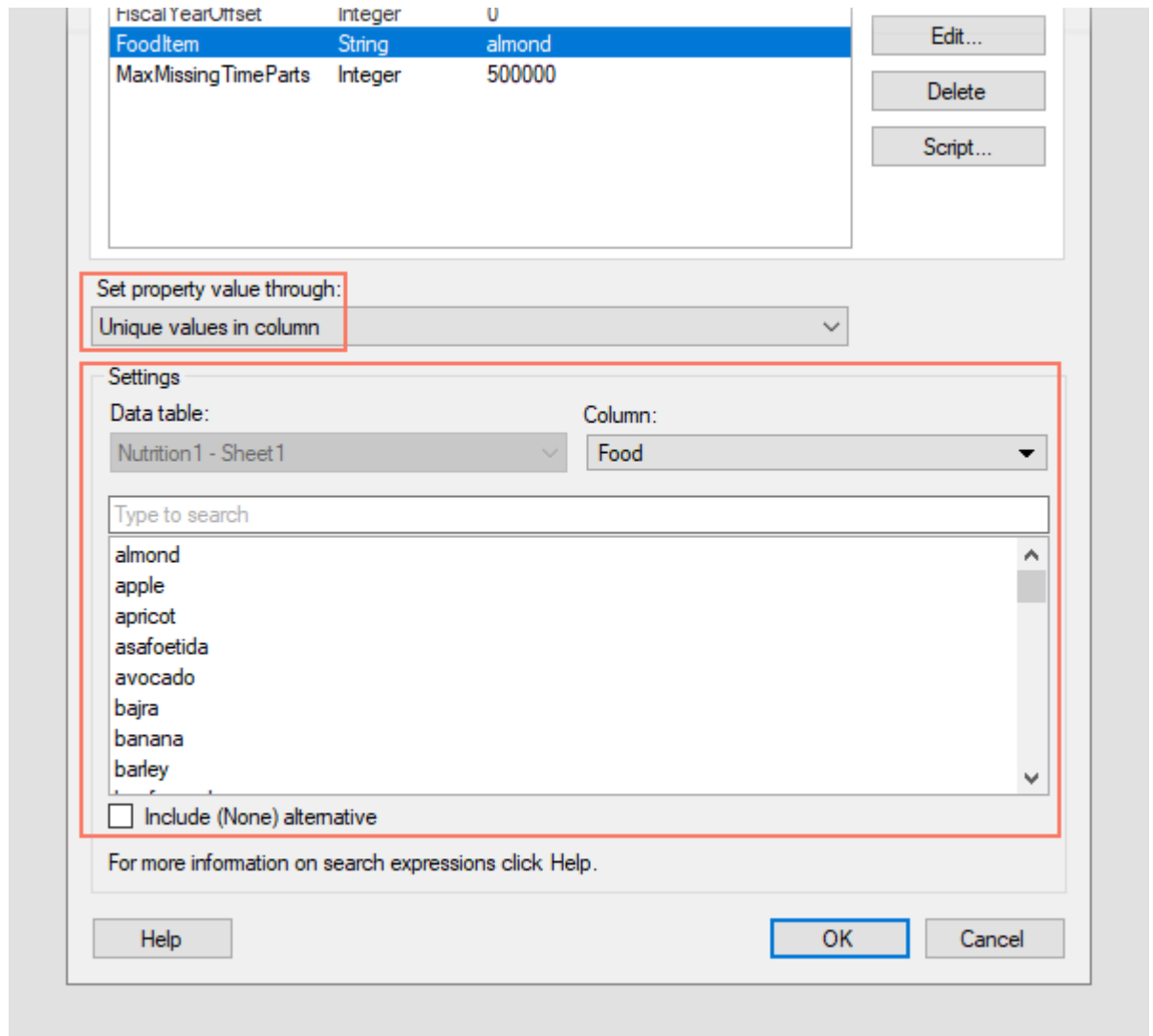
1. In a text area, [add the property control](#), in this case, a drop-down list.



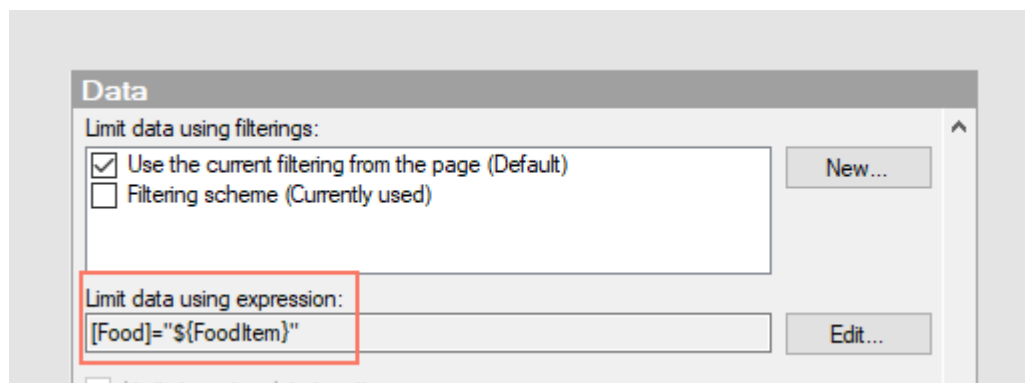
2. In the Property Control dialog, create the property via **New**. Give it a name, specify its data type and a start value. In this example, we call the property 'FoodItem'.



- When the property exists and has been selected, define which values the property should be able to have. In this example, the unique values in the 'Food' column should be available for selection in the drop-down list.



- Finally, connect the property to the **Limit data using expression** setting on the **Data** page of the Properties dialog (via **Edit**). The expression excludes all other food types except the value of the property as currently selected in the drop-down list.



## Using a property in a calculated expression

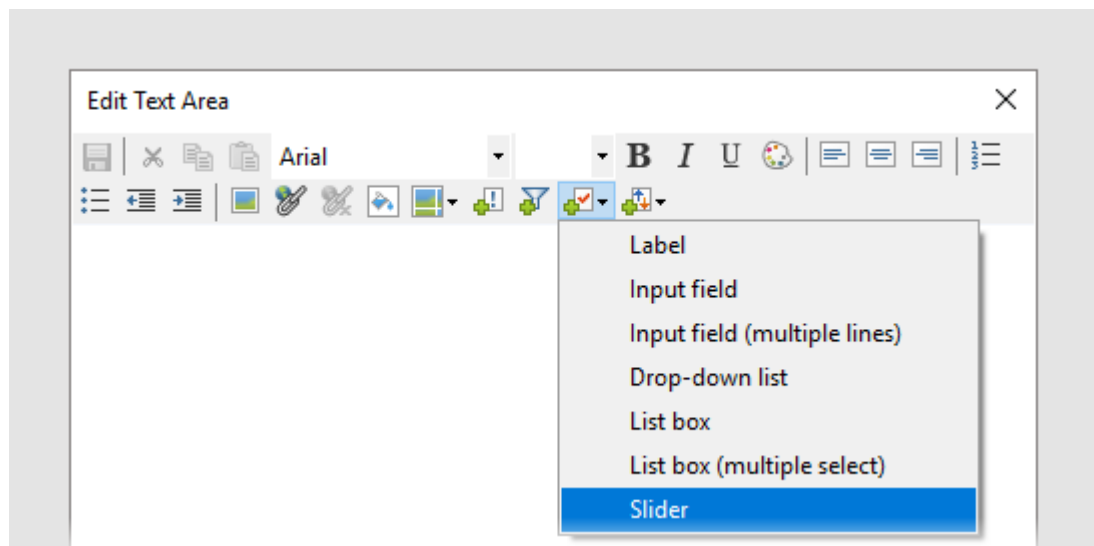
In this example, assume you are trading with foreign countries, and the figures you want to visualize are dependent on the current exchange rates. Then, you can create a property that can assume different exchange rates and include this property in the conversion calculations. Anyone using the analysis can then simply adjust the value of the property to reflect the current exchange rate, and automatic recalculations will take place immediately. Both the property creation and the property value modifications can be handled from a text area.

### Prerequisites

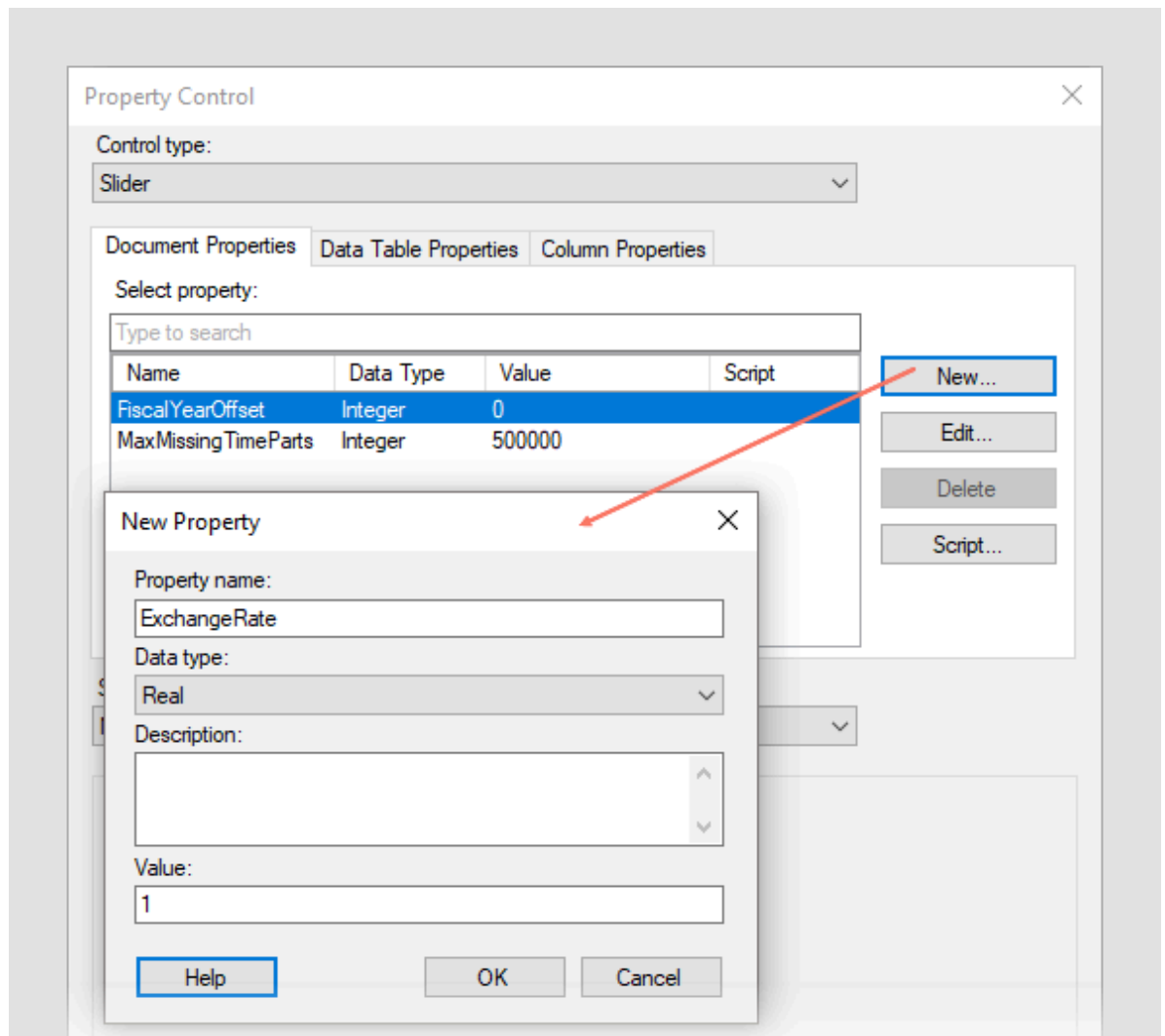
The functionality described here must be authored using the installed client.

### Procedure

1. Create or edit a text area.
2. In edit mode, [add the property control](#) to the text area. In this example, we add a **Slider**.

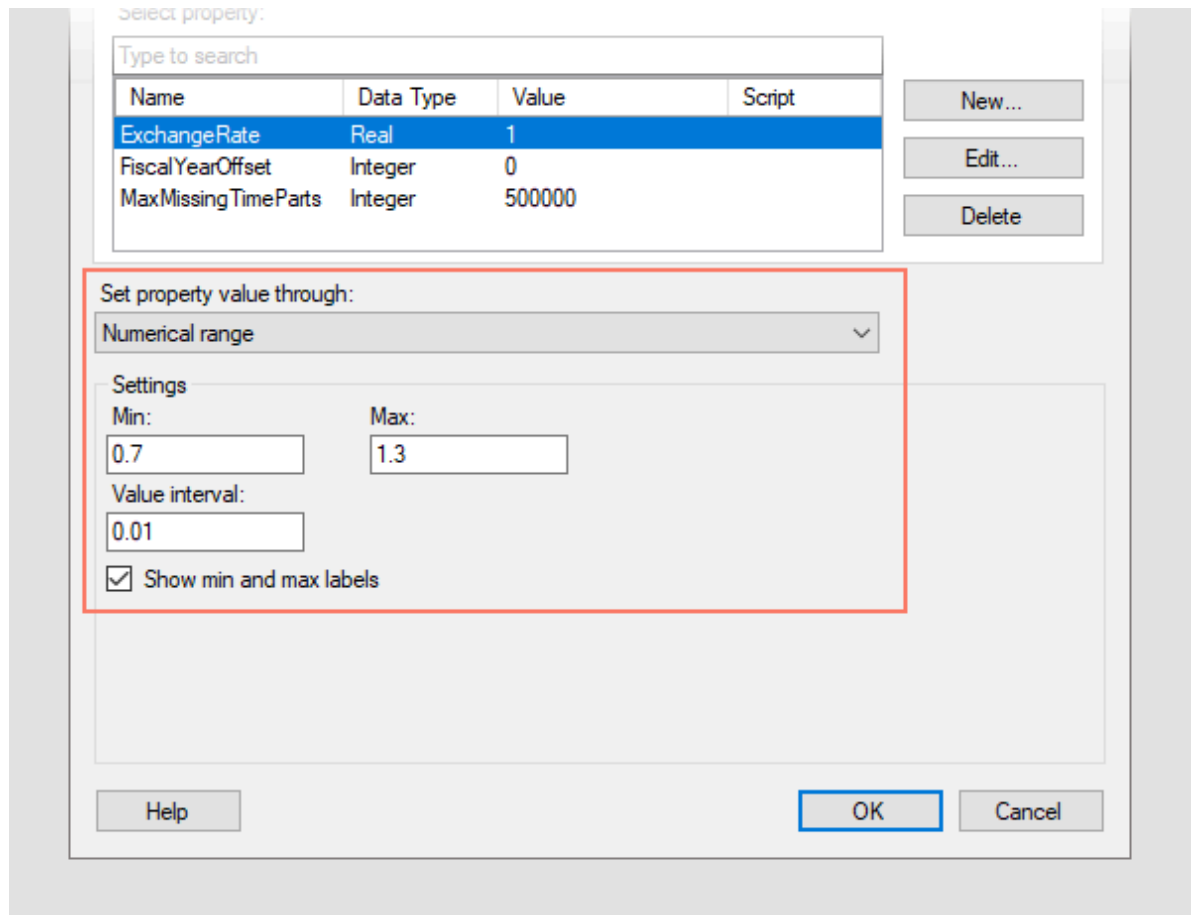


3. In the Property Control dialog, create the property via **New**. Give it a name, specify its data type, and a starting value. In this example, we call the property 'ExchangeRate'.

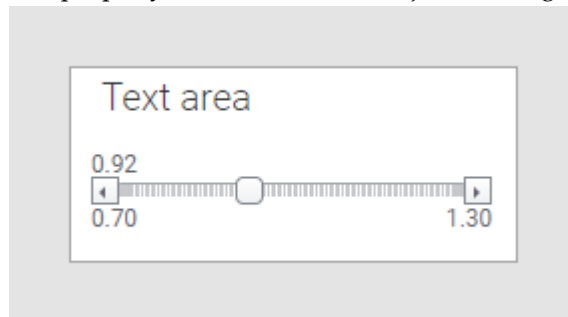




- When the property exists and has been selected, you can adjust the appearance of the slider control at the bottom of the Property Control dialog. The total range is defined, as well as the stepwise intervals when dragging the slider.



- Click **OK** when you are done, and **Save** the text area. The property value can now be adjusted using the slider.



## Result

You can now use the property to convert the dollar amounts in the data table below to EUR, and add them as a new data column. Select **Data > Add calculated column** to open the Add calculated column dialog, and enter `{ExchangeRate} * [Amount]` in the **Expression** field (note the syntax).

Now you can drag the slider to the current exchange rate, and the amounts in the added column are automatically converted using the specified exchange rate.

Customer ID	Amount		Customer ID	Amount	$\$(\text{ExchangeRate}) * [\text{Amount}]$
SSMM55001	\$6999		SSMM55001	\$6999	\$6439
SSMM55002	\$7156		SSMM55002	\$7156	\$6584
SSMM55003	\$731		SSMM55003	\$731	\$673
SSMM55004	\$1431		SSMM55004	\$1431	\$1317
SSMM55005	\$5429		SSMM55005	\$5429	\$4995
SSMM55006	\$467		SSMM55006	\$467	\$430
SSMM55007	\$467		SSMM55007	\$467	\$430
SSMM55008	\$0		SSMM55008	\$0	\$0

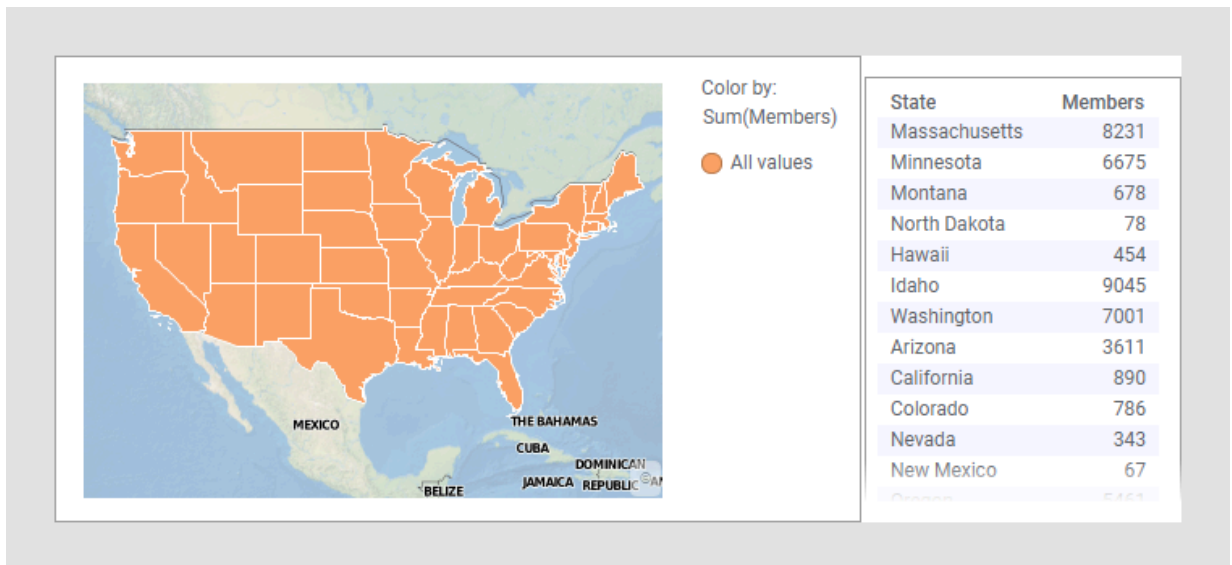
### Using a property for coloring purposes

The **Color by** axis in the map chart feature layer below is set to the Members column in the data table to the right. All states have the same color so far, because the Color mode is set to Fixed. Assume you wish to identify states where the number of members exceeds a certain user-controlled value that you specify in a text area. For example, you might want to color all states that have more than 1000 members in green.

#### Prerequisites

The functionality described here must be authored using the installed client.

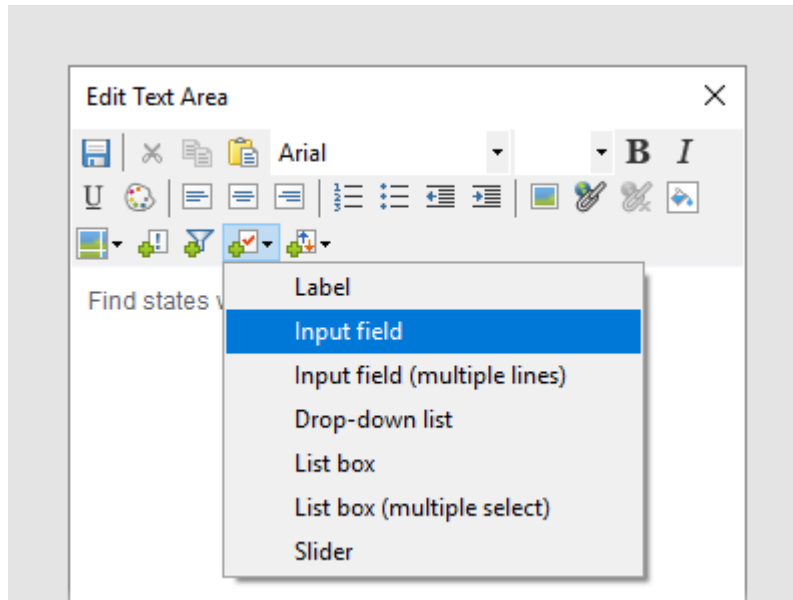
To do this, a color rule is added to the Fixed color mode. The rule includes a property, whose value is controlled from a text area:



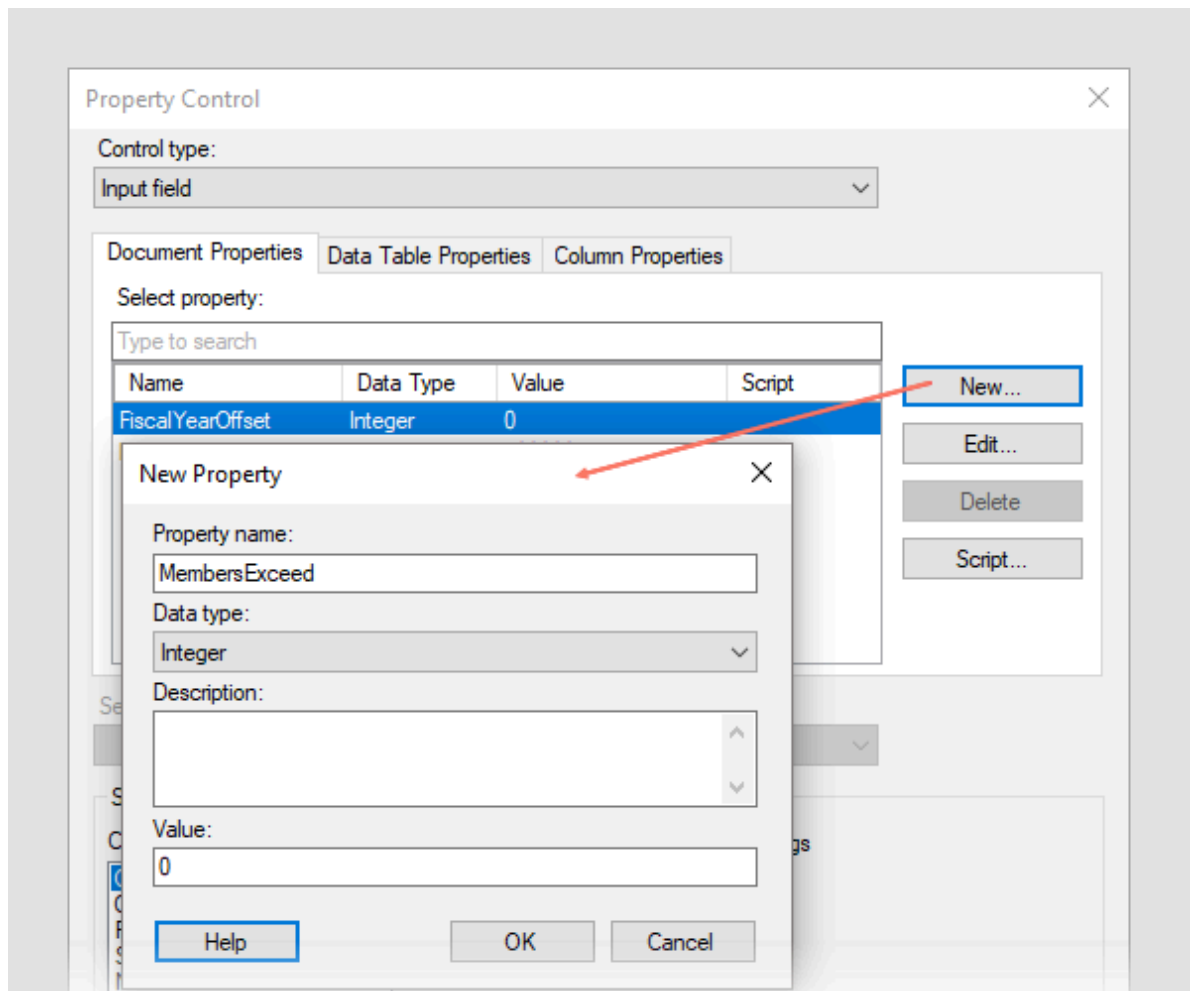
#### Procedure

1. Create or edit a text area.

2. In edit mode, [add the property control](#) to the text area. In this example, we add an **Input field**.

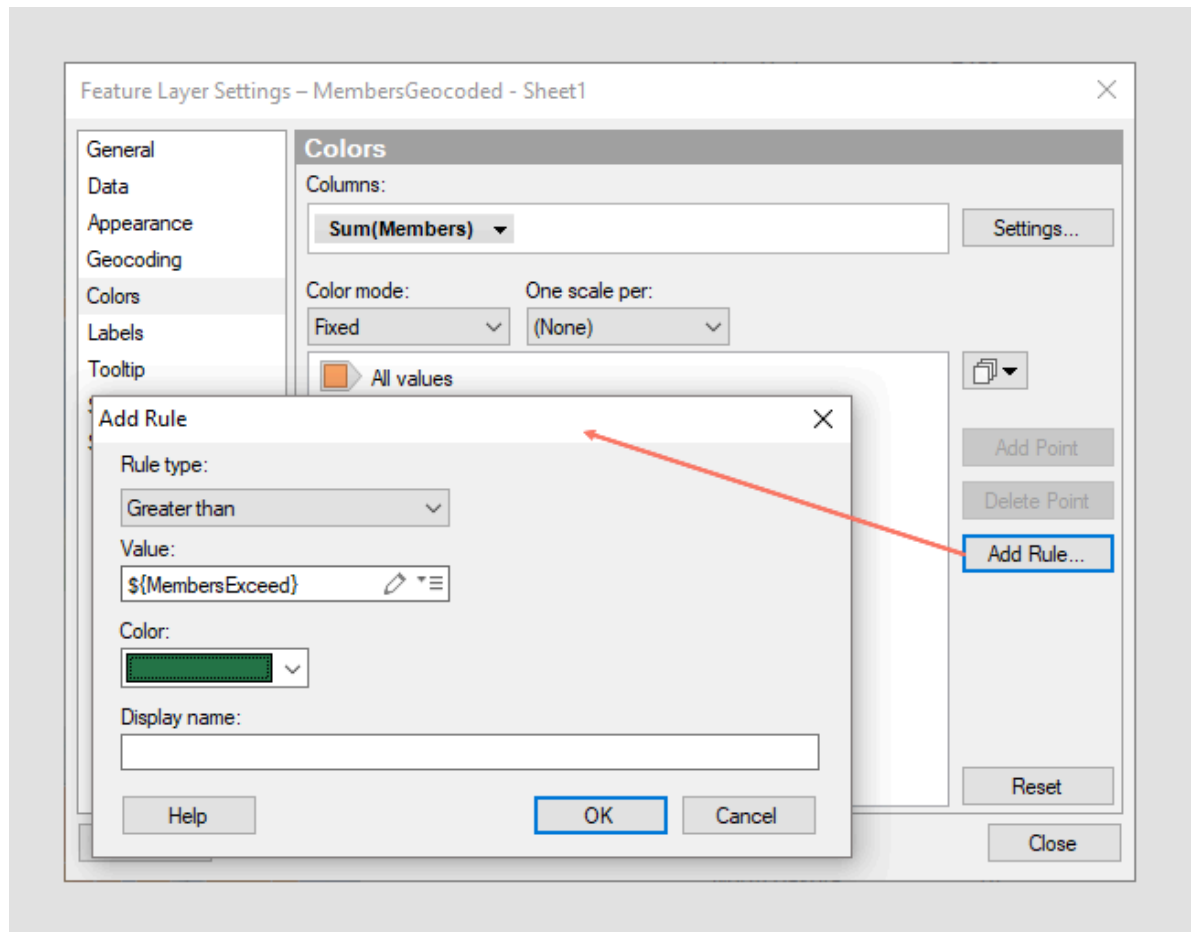


3. In the Property Control dialog, create the property via **New**. Give it a name, specify its data type, and a starting value. In this example, we call the property 'MembersExceed'.



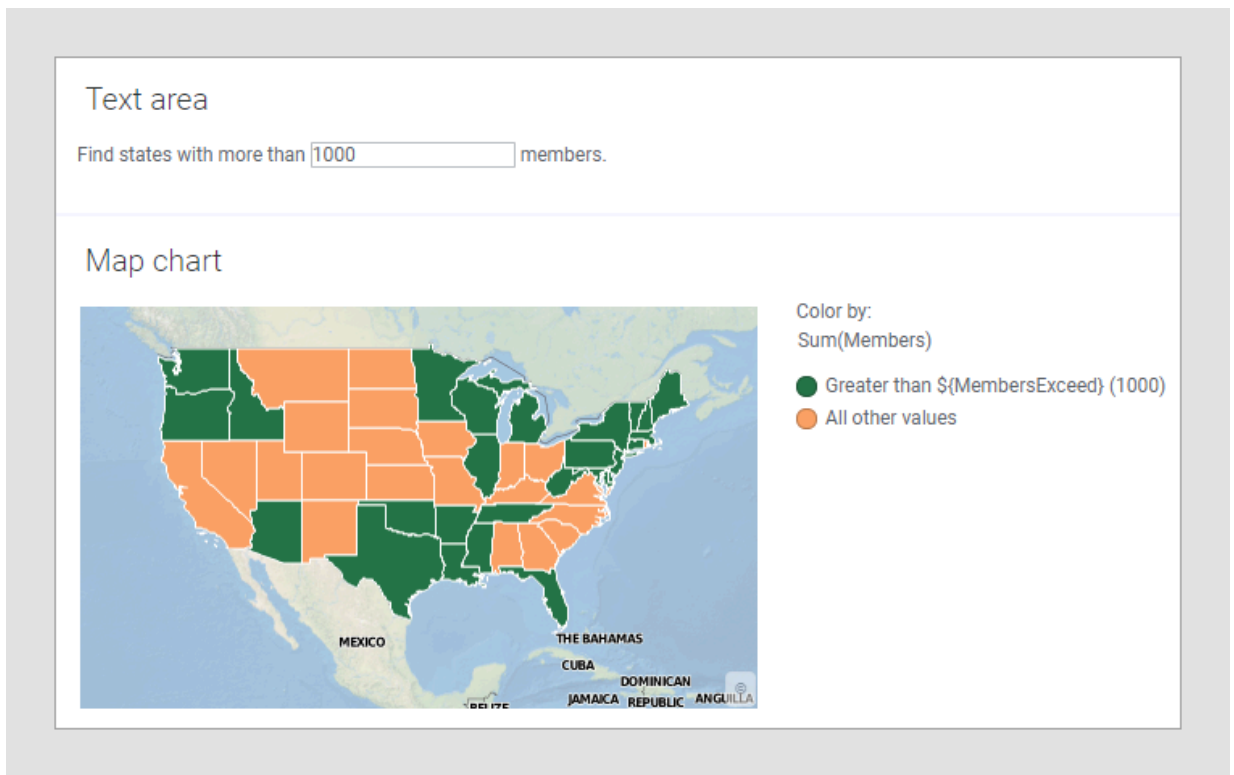
No other settings are needed within the Property Control dialog.

- Now you can connect the property to the place that should react on changes to the property value. In this case it is a color rule on the **Colors** page in the Feature Layer Settings dialog. Click **Add Rule**, set **Rule type** to **Greater than**, and in the **Value** field, select **Custom expression**, and enter the created property `${MembersExceed}` (note the syntax) below **Expression**.



## Result

Now you can type a number in the input field in the text area, and the map chart coloring updates according to the color rule.



### Adding lines using a property

You can use a property as the input to a line or a curve. A property value can either be a string expression in itself, or it can be used as a variable in a custom expression.

#### Prerequisites

The functionality described here must be authored using the installed client.

## Using a property value to define a line in a visualization

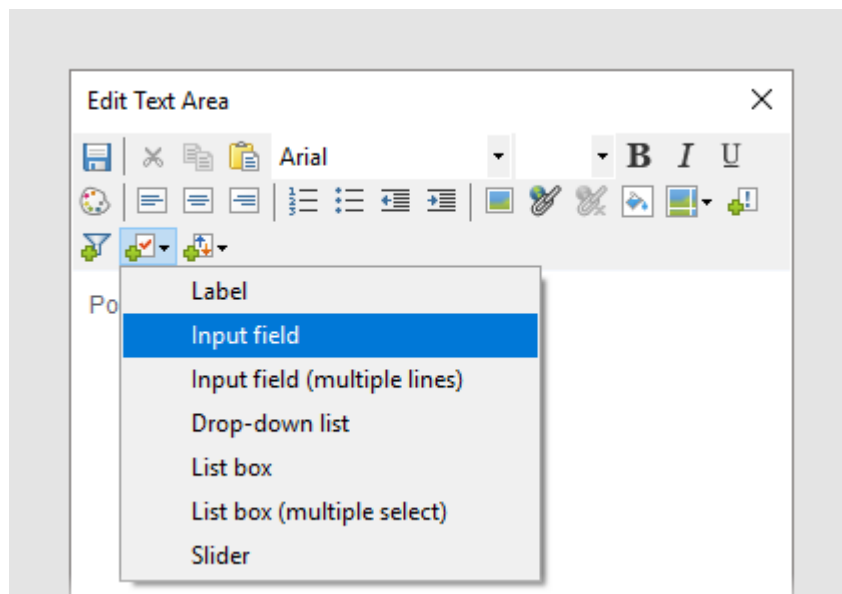
The first example describes how the position of a horizontal line in a scatter plot can be changed by anyone using a property control in a text area.



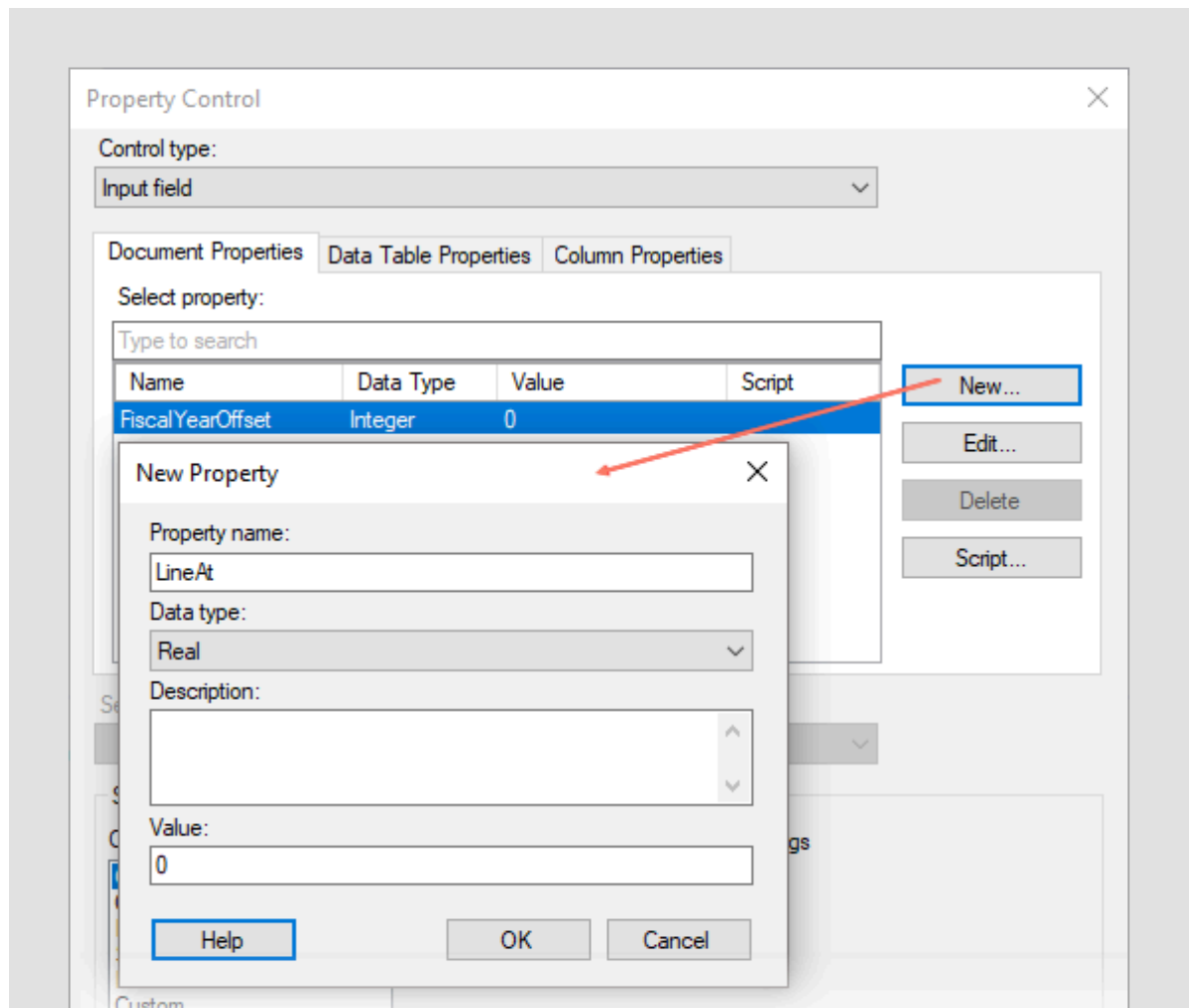
Follow the steps to create this type of analysis.

### Procedure

1. Create or edit a text area.
2. In edit mode, [add the property control](#) to the text area. In this example, we add an input field.



3. In the Property Control dialog, create the property via **New**. Give it a name, specify its data type, and a starting value. In this example, we call the property 'LineAt'.



4. When the property exists and has been selected, you can adjust the formatting of the values at the bottom of the Property Control dialog. Click **OK** when you are done, and **Save** the text area.
5. The next step is to connect the created property to the setting that should be affected by the property value. In the visualization where you want to add the line, right-click and select **Properties**.
6. Click **Lines & Curves**.
7. Click **Add > Horizontal Line > Straight Line**.

8. In the Horizontal Line dialog, select the **Custom expression** option, click **Edit** and, in the Custom Expression dialog, enter `${LineAt}`.

This is the syntax used when adding the value of the property as text in an expression (and also the default if you double-click the property in the **Available properties for column** list). See [Properties in expressions](#) on page 907 for more information regarding the different syntaxes.

The screenshot shows the 'Horizontal Line' dialog box. Under the 'Line position' section, the 'Custom expression' radio button is selected and highlighted with a red rectangle. The text field next to it contains the expression `${LineAt}`. To the right of this field is an 'Edit...' button. Below this section is a checkbox for 'Use axis transform in line calculation' which is unchecked. In the 'Line name' section, the 'Automatic' radio button is selected. At the bottom of the dialog are three buttons: 'Help', 'OK' (which is highlighted with a blue border), and 'Cancel'.

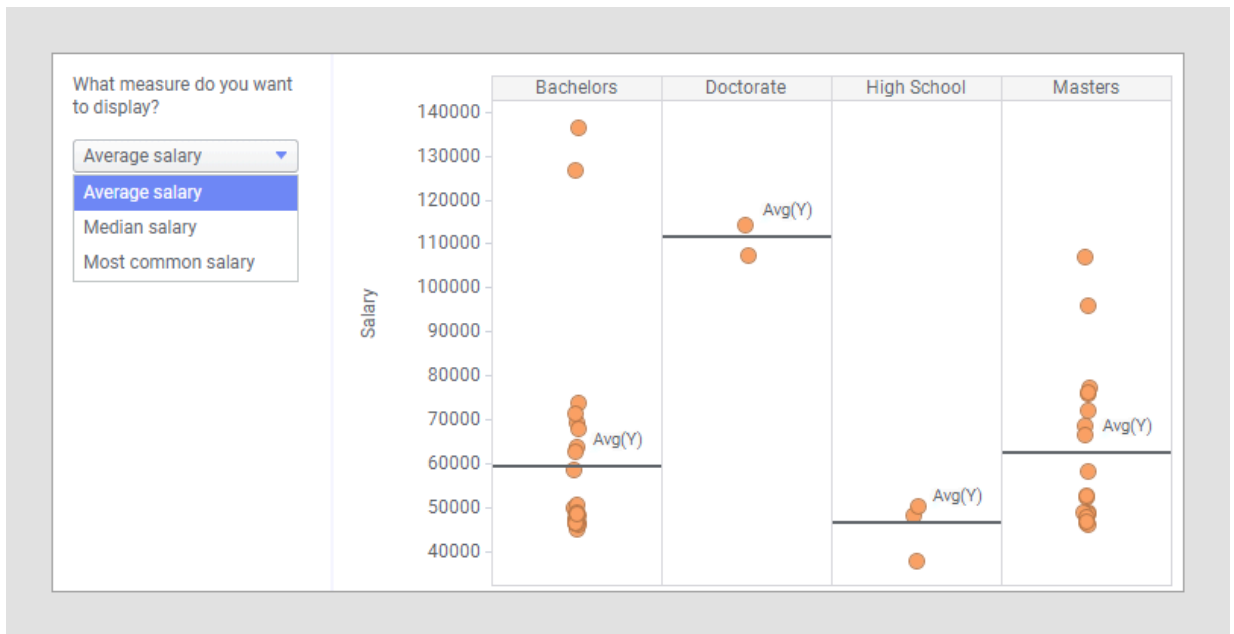
9. Click **OK**.  
The curve is shown in the visualization and you can use the property control in the text area to change the value, and hence, the position of the line.

### Using a property expression to define a line showing different statistical measures

In this example, a scatter plot is used to show salaries for employees, trellised per education level. The text area contains a property control in the form of a drop-down list, where a dashboard user can select



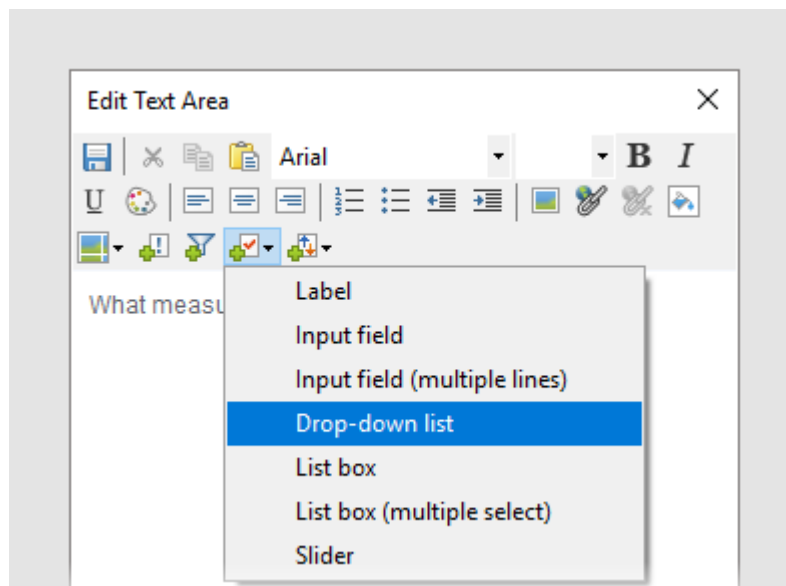
to add horizontal lines that show either the average, the median, or the most common salary within each education level.



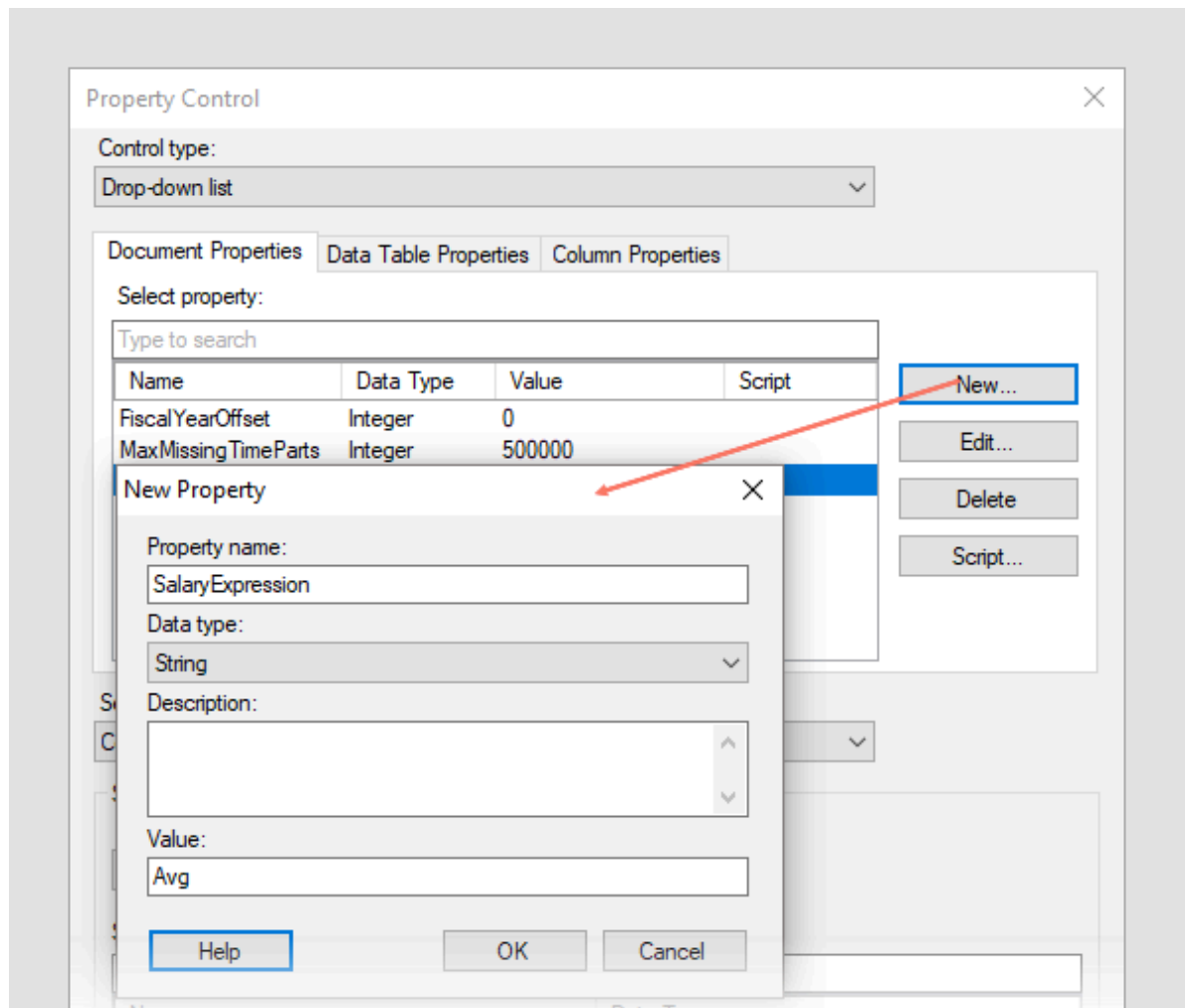
Follow the steps to create this type of analysis.

### Procedure

1. Create or edit a text area.
2. In edit mode, [add the property control](#) to the text area. In this example, we add a drop-down list.



3. In the Property Control dialog, create the property via **New**. Give it a name, specify its data type, and a starting value. In this example, we call the property 'SalaryExpression'.



4. Now that the property exists, define which values to include in the drop-down list. In this example, you define **Fixed values** yourself, by typing the text to show under **Display Name**, and by adding their values to the right.

SalaryExpression String Avg

Delete

Script...

Set property value through:

Fixed values

Settings

Display Name	Value
Average salary	Avg
Median salary	Median
Most common salary	MostCommon

Add

Remove

Move Up

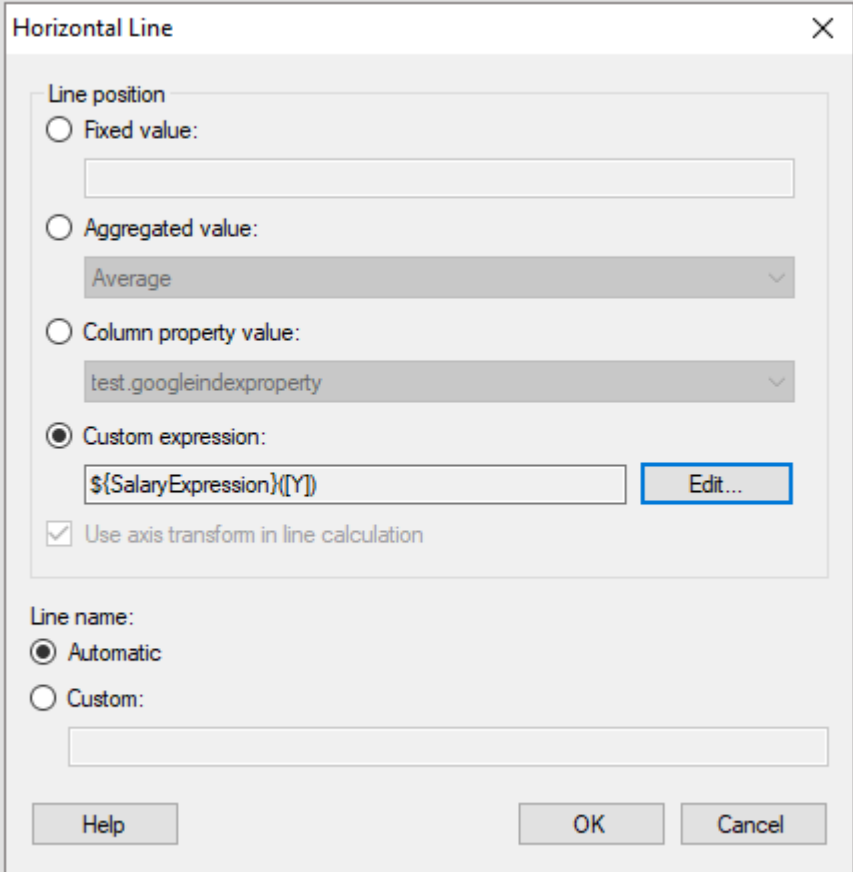
Move Down

☐ Include (None) alternative

Help OK Cancel

5. In the visualization where you want to add the curve, locate the **Properties**.
6. Click **Lines & Curves**.
7. Click **Add > Horizontal Line > Straight Line**.
8. In the Horizontal Line dialog, select the **Custom expression** option, click **Edit** and, in the Custom Expression dialog, enter `${SalaryExpression}([Y])`, where `SalaryExpression` is the property

you have created (the result of the custom expression  $\${SalaryExpression}([Y])$ ) will, depending on the selection in the drop-down list, become  $Avg([Y])$ ,  $Sum([Y])$ , or  $MostCommon([Y])$ .



**Horizontal Line** [X]

**Line position**

☐ Fixed value:

☐ Aggregated value:

Average

☐ Column property value:

test.googleindexproperty

☒ Custom expression:

$\${SalaryExpression}([Y])$  [Edit...](#)

☒ Use axis transform in line calculation

**Line name:**

☒ Automatic

☐ Custom:

[Help](#) [OK](#) [Cancel](#)

9. Click **OK**.

The curve is shown in the visualization and you can select where a line should be drawn in the visualization by making a choice in the drop-down list. For example, in the image below, Median salary is selected.



## Using a property expression to define a curve

A property can either be a string expression in itself, or it can be used as a variable in a custom expression. This step instruction assumes that the property contains a string which is an expression.

### Procedure

1. For the visualization where you want to add the curve, open the visualization properties.
2. Locate the **Lines & Curves** section.
3. Click **Add** and select **Curve Draw**.
4. In the Curve Draw dialog, call the property containing the expression in the **Curve expression** field using the Text syntax: For example, `${curve.expression}`.

The document property `curve.expression` could in this example hold a value such as  $2+3*x$ . The expression could also contain other properties. For example, the value of `curve.expression` could be `2+DocumentProperty("ExpressionConstant")*x`, where `ExpressionConstant` is another property which holds a numeric value. Note that the syntax for calling a property differs when calling it to retrieve the value of the property. See [Properties in expressions](#) on page 907 for more information regarding the different syntaxes.

5. Click **OK**.

## Understanding properties in a Spotfire analysis


As an analysis author, you can investigate the metadata and learn more about the columns and data tables in your analysis. In the data canvas, you can inspect both automatically created and custom column and data table properties. If you add custom properties, you should also add a description to help others understand what the property is used for.

### Column properties

The following column properties are often available by default:

Property	Description
Name	<p>The name is a trimmed and unique identifier for the column. It is used as the display name for the column in the user interface.</p> <p>You can either <a href="#">change the name</a> in the <b>Data in analysis</b> flyout or by <a href="#">editing the column property value</a> in the <b>Data canvas</b>.</p>

Property	Description
ColumnType	<p>The type of column.</p> <p>The following are the mutually exclusive column types:</p> <p>None - no column type defined.</p> <p>Imported - a standard column opened from a file, the clipboard, a data connection or from an information link.</p> <p>Calculated - a column created using the <a href="#">Add calculated column</a> tool.</p> <p>Binned - a column created from another column using the Add binned column tool.</p> <p>Frozen - a Calculated/Binned/Tags/Result column that has been frozen and behaves like an imported column.</p> <p>Result - an output result from a calculation such as a clustering.</p> <p>Tags - a column displaying the used tags of a tag collection.</p> <p>Mask - a temporary column created using the <b>Edit &gt; Marked rows &gt; Filter to</b> command. This command creates a column with information of whether or not a column was filtered to, the last time the command was applied. There can only be one mask column in each data table.</p> <p>Hierarchy - a predefined hierarchy set up using the <a href="#">Add hierarchy</a> tool.</p>
DataType	<p>The type of the values in the column.</p> <p>The data type of a column cannot be changed directly as a property, but you can usually change it from the expanded <b>Data in analysis</b> flyout, as described in <a href="#">Changing the data type of a column</a> on page 158.</p> <p>If this approach for some reason does not work, you can also try to <a href="#">replace the data table</a> with itself and add a <a href="#">transformation step</a> (installed client only) to change the data type.</p>
ExternalName	The name of the column in the original data source or view. The purpose of this property is to allow column matching for linked data and add/replace data operations.
ExternalId	The globally unique id of the column, if available from the data source. Available for many columns imported via information links, but not all. The property is invalid if the column has no external id. For data connections the external ID contains both the name of the original column in the database and the name of the original data table in the database.
IsValid	True if the column expression is valid. The property itself is valid only for calculated and binned columns, and the property value is false if there is a dangling reference in the column expression.
Expression	For calculated columns only. Displays the expression used to calculate the column.
DerivedExpression	For calculated columns only. Displays the expression used to calculate the column after the pre-processor values have been evaluated.
Origin	A text string that describes where the column comes from. Typically set by a data source, or by a tool for a result column.
Description	A description of the column entered by an author as an annotation.
LinkTemplate	A link template used to create links to a website from the values in a table. For example, the link template can be "http://www.{\$.}com", where "{\$.}" represents the data value from the column. See also <a href="#">Editing data table or column property values</a> on page 247.
DefaultContinuousColorScheme	The name of the default continuous color scheme that will be used when continuous coloring is applied to this column. If no color scheme is specified, or if the specified color scheme is not available, the Spotfire Continuous color scheme will be used.

Property	Description
DefaultCategoricalColorScheme	The name of the default categorical color scheme that will be used when categorical coloring is applied to this column. If no color scheme is specified, or the specified color scheme is not available, the Spotfire Categorical color scheme will be used.
ContentType	<p>Each column can have a specified content type. Renderers use this property as input to know what to display. Use the form toplevel/subtype, for example, text/plain or image/jpg. For Geometry columns the content type should be set to application/x-wkb if you want to show the geometry information as images. If you are using Spotfire to display chemical structures from an SDF file then the content type should be set to chemical/x-mdl-molfile for the molfile column.</p> <p> Do not use a space when specifying the content type.</p>
Keywords	Keywords defined on the column element can be used when searching for columns with the following syntax: <code>Keywords:&lt;desired keyword&gt;</code> . For example, <code>Keywords:Sales</code> .
GeocodingType	<a href="#">Geocoding</a> properties are used when a column can be used to position geographical data on a map. This property specifies the type of geocoding information contained in the column, such as State or City.
GeocodingRepresentation	<a href="#">Geocoding</a> properties are used when a column can be used to position geographical data on a map. This property specifies how the geocoding information contained in the column is represented. This can be a language code such as en-us or a standard identifier such as ISO3166-2:us.
GeocodingHierarchyName	<a href="#">Geocoding</a> properties are used when a column can be used to position geographical data on a map. This property specifies the name of the geocoding hierarchy that the column belongs to.
PreferredAggregationMethod	The name of an aggregation method that is used by the heuristics when an aggregated expression is created from this column. You can change the preferred aggregation method from <a href="#">the expanded Data in analysis flyout</a> .
ValidInsightPredictor	Indicates if this column should be included when calculating recommendations.

## Data table properties

The following data table properties are sometimes available by default:

Property	Description
Keywords	<p>Keywords for the data table.</p> <p>When data tables have been added using information links, the keywords defined in <b>Information Designer</b> will be used.</p>
Description	<p>A description of the data table.</p> <p>When data tables have been added using information links, the description defined in <b>Information Designer</b> will be used.</p>
ExternalId	The globally unique identifier of the data table in the original data source, if available.
MapChart.GeometryType	A string value that defines the type of feature when a geocoding table contains a Geometry (shape) column. The value can be either Point, Line or Polygon.
MapChart.IsGeocodingTable	Specifies if the data table is a geocoding table. True or False.

Property	Description
MapChart. GeocodingHierarchyName	A string value used to group geocoding tables that belong in the same hierarchy.
MapChart. GeocodingHierarchyPriority	An integer value between 0 and 100 (100 being the highest priority) used to control the priority of the hierarchy in which the table is included. A geocoding hierarchy with higher priority will be selected first if two hierarchies result in an equal number of column matches when automatic geocoding is applied.
MapChart.GeocodingAutoload	Specifies if a geocoding table should be allowed to be automatically downloaded to the analysis when Spotfire finds out that it can be used as a geocoding table for the current data table. This value can be used to prevent large geocoding tables from being automatically downloaded. This property is important when geocoding tables are saved in the library. True or False.
MapChart. GeocodingHierarchyVersion	Specifies the version of the geocoding hierarchy in the form of a date (YYYYMMDD).
MapChart.GeographicCrS	Specifies the coordinate reference system describing the data (e.g., "EPSG:4326" for WGS 84).
MapChart.IsGeocodingEnabled	Specifies if the geocoding table is currently enabled. This property is important when geocoding tables are saved in the library. True or False.
MapChart.ExternalGeographicCrS	Specifies the EPSG code of the projection identified by the shapefile projection web service. In use when the identified projection already exists in the Spotfire list of projections.
MapChart. ExternalGeographicProj4Def	Specifies the Proj4 definition generated by the shapefile projection web service. The value stored in this property is used by Spotfire to create a Generated CRS. In use when the shapefile's projection does not already exist in the Spotfire list of projections.

For more information about geocoding data tables, see [Geographic location and geocoding](#) on page 169 and [Specifying new geocoding tables](#) on page 172.

## Document properties

The best overview of document properties is in the Document properties panel, **View > Document properties**. See also [Adding or editing a document property](#) on page 245.

The following document properties are often available by default:

Property	Description
MaxMissingTimeParts	Specifies the maximum number of missing time parts that should be allowed to be replaced using the <b>Compensate for missing values</b> setting, available in the visualization properties of some visualizations. See <a href="#">Additional operations with time hierarchies</a> on page 510 for more information.
FiscalYearOffset	Specifies the number of months from the start of the calendar year to the start of the fiscal year. For instance, a value of -1 specifies that the fiscal year starts in December of the previous calendar year, whereas a value of 2 specifies that it starts in March of the current calendar year. If you use any of the <a href="#">Date and time functions</a> on page 866 <code>FiscalMonth</code> , <code>FiscalQuarter</code> or <code>FiscalYear</code> in expressions and you leave out the optional second argument, then the value of this property will be used instead.



# Visualizations

---

There are a number of different visualization types available in Spotfire, and which to choose depends on your data and what you want to show. If you cannot find the type of visualization you are looking for directly, you can also look for visualization mods on the Community, or create your own mod.

See [Developing mods](#) on page 630 for more information about visualization mods.

All of the visualizations you add can be modified by changing their configuration. See the corresponding section for the different visualization types for some examples and configurations you might want to use.



A few of the visualization types described here can only be authored using the installed client. Once added to an analysis, all visualizations except for the 3D scatter plot can be viewed in the web client.

## Creating a visualization


---

Data can be of many different kinds. Therefore, to present the data in the best way, there are several types of visualizations available.

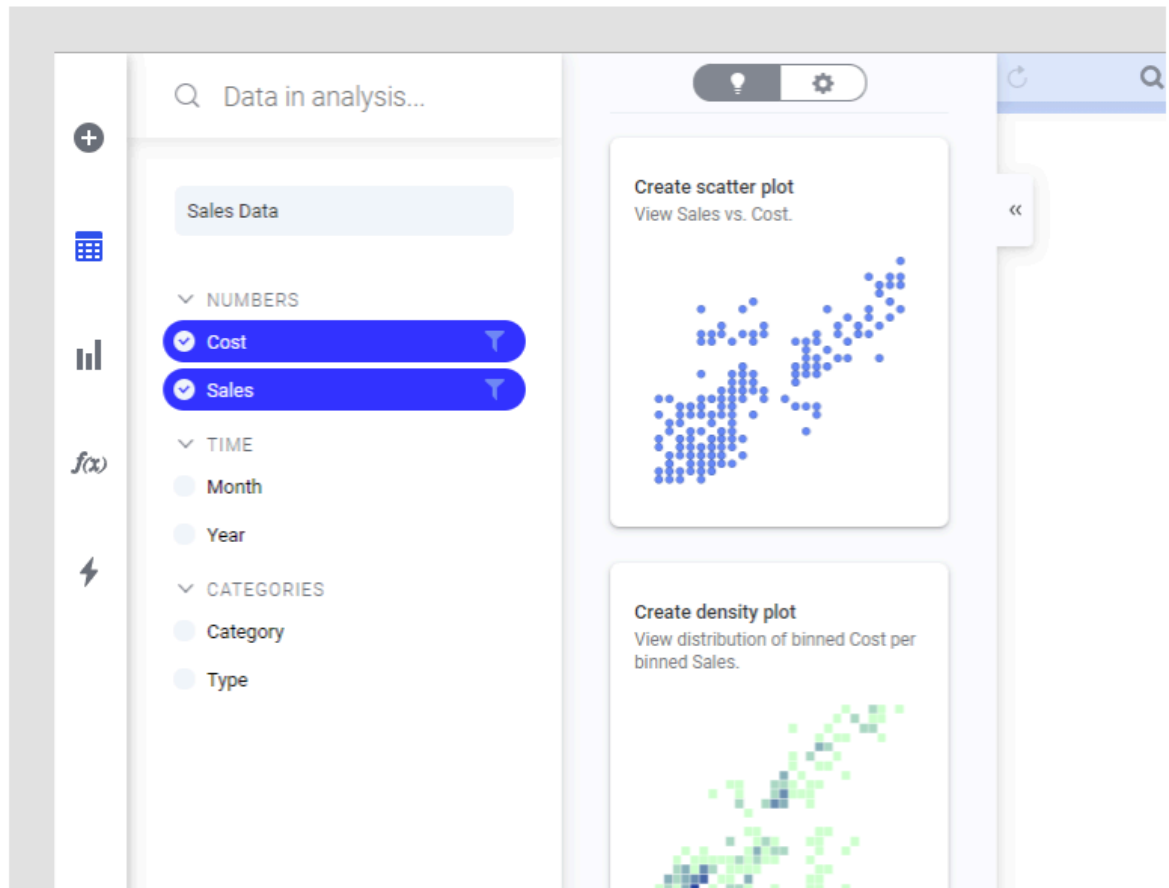
### Create a visualization from the Data in analysis flyout

To create visualizations, use the **Data in analysis** or **Visualization types** flyout, or search for what you want to visualize. A page in an analysis can contain more than one visualization, but to make room for further visualizations, you can add more pages.

When you want to explore certain data columns in your data table, and get suggestions on how they can be visualized, you use the **Data in analysis** flyout to create visualizations.

1. On the authoring bar, click  to open the **Data in analysis** flyout.
2. If your analysis contains more than one data table, select the data table you are interested in from the drop-down list.

3. Select the data columns of interest.




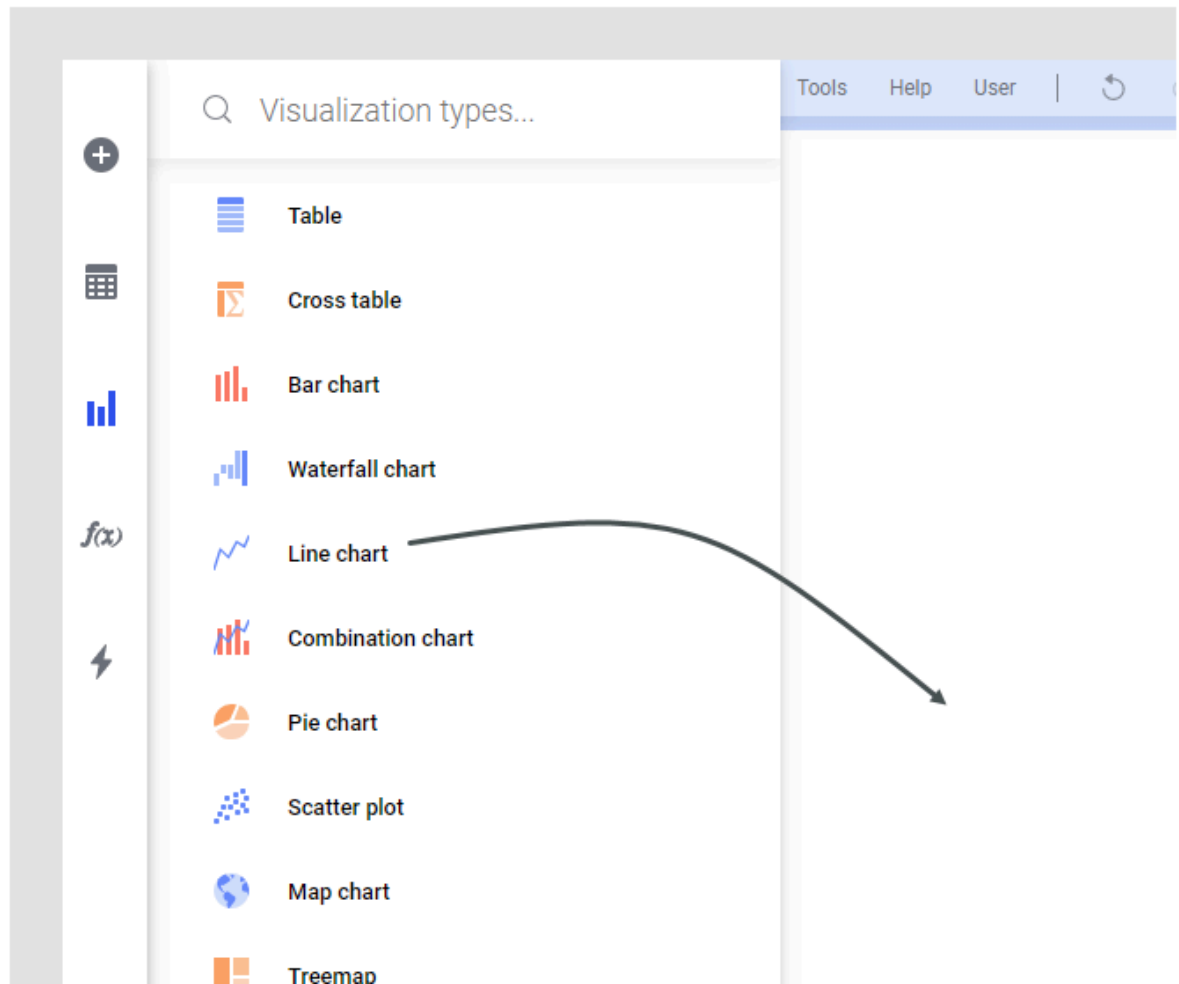
To the right, you get suggestions on visualizations based on the selected data. If you hover with the mouse on a visualization, you can click **MORE LIKE THIS**, and similar visualizations are suggested. The visualizations might also show data from other columns, if certain findings related to the selected data are discovered.

4. Click, or drag, the visualization you find suitable to the analysis page.
5. Explore it as it looks, or adjust it to your needs.

### Create a visualization from the Visualization types flyout

Another option is to create visualizations from the very beginning and make your own settings. To learn how to set up the different visualizations from scratch, click the links at the bottom of this page.


1. On the authoring bar, click  to open the **Visualization types** flyout.



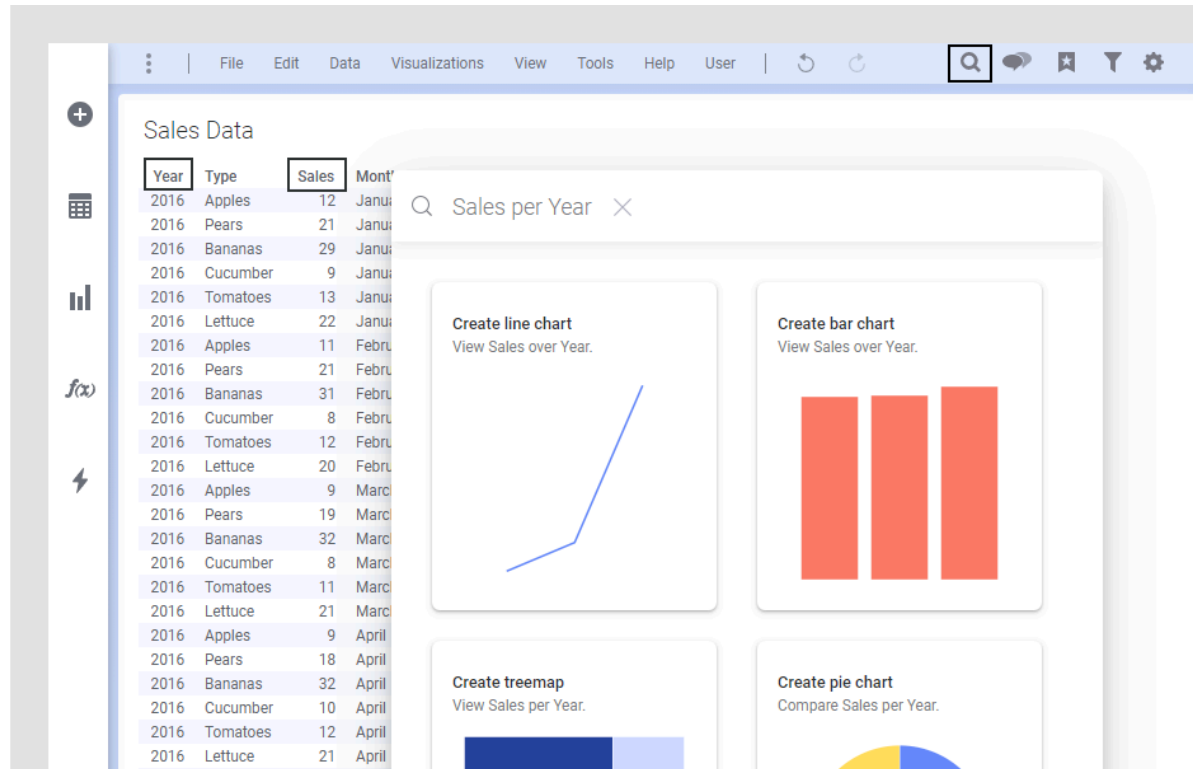
2. Click, or drag, the visualization type you want to use to the analysis page.
3. Adjust it to your needs. For information on the different visualization types, see the links at the bottom of this page.

### Create a visualization by searching

You can create a visualization by typing search criteria in a text field.

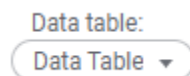
1. Click **Find**  on the menu bar.

2. In the opened search field, type what you are looking for in your data.



You get suggestions on visualizations matching your search criteria. The visualizations might also show data from other columns, if certain findings related to the selected data are discovered.

3. Drag the visualization you want to use to the analysis page
4. Adjust the visualization to your needs. For example, you can change which data table to base the visualization on using the data table selector in the legend or in the **Data** section of the visualization properties.



You can also change the axes, the colors or update other properties. For information on the different visualization types, see each section.



Some visualization types can only be authored using the installed client.

## Table

The table in Spotfire works much like any other table you might be familiar with. It presents the data as a table of rows and columns, and is used to see details and compare values.

By clicking a row you mark it, and by dragging the mouse pointer over several rows you can mark more than one row. The appearance of the table can be changed. For example, the [order of the columns](#) and their [widths](#) can be adjusted, [headers](#) can be displayed horizontally or vertically, and the [alignment of the texts](#) can be specified.

You can [sort the rows](#) in the table according to different columns, or filter out unwanted rows by using the filters.

Year	Month	Category	Type	Sales	Cost
2019	January	Fruit	Apples	12	10
2019	January	Fruit	Pears	21	13
2019	January	Fruit	Bananas	29	26
2019	January	Vegetables	Cucumber	9	6
2019	January	Vegetables	Tomatoes	13	11
2019	January	Vegetables	Lettuce	22	20
2019	February	Fruit	Apples	11	9
2019	February	Fruit	Pears	21	14
2019	February	Fruit	Bananas	31	27
2019	February	Vegetables	Cucumber	8	6
2019	February	Vegetables	Tomatoes	12	10
2019	February	Vegetables	Lettuce	20	19
2019	March	Fruit	Apples	9	10
2019	March	Fruit	Pears	19	13
2019	March	Fruit	Bananas	32	27
2019	March	Vegetables	Cucumber	8	6
2019	March	Vegetables	Tomatoes	11	11



Press the Shift key and click on a column header to sort by that column. Press Shift and click again to change sort order. To sort by an additional column, you can press Ctrl+Shift and click on the column header of interest. Press Ctrl+Shift and click again to change sort order.

All visualizations can be configured to show data limited by one or more markings in other visualizations only (details visualizations). Tables can also be limited by one or more filterings. You can also configure a table without any filtering at all. You can also set up a table without any filtering at all. See [Adding data limitations for a visualization](#) for more information.

### Working with in-database data in table visualizations

When you work with in-database data from relational data sources, the table visualization can only show a limited number of rows. The default value of this limit is 10 000 rows. For data that exceeds this limit, only the first 10 000 rows will be visualized in the table. A Spotfire administrator can configure the row limit on your Spotfire server, by editing the preference `TableVisualizationExternalRowLimit` in the Administration Manager. A primary key must be specified to enable highlighting and markings in the table visualization.




For in-db data from cube data sources you need to manually specify which columns to show in the table. Choosing a small number of columns helps reducing the size of the resulting query and improves performance.

## Creating a table

The table visualization presents all details of the loaded data. The individual values are arranged in columns and rows.

### Procedure

1. On the authoring bar, click  to open the **Visualization types** flyout.
2. Drag the **Table** visualization type to the wanted position on the analysis page.

## Example

Assume you load the Microsoft® Excel file below.

	A	B	C	D	E	F
1	Year	Month	Category	Type	Sales	Cost
2	2020	January	Fruit	Apples	12	10
3	2020	January	Fruit	Pears	21	13
4	2020	January	Fruit	Bananas	29	26
5	2020	January	Vegetables	Cucumber	9	6
6	2020	January	Vegetables	Tomatoes	13	11
7	2020	January	Vegetables	Lettuce	22	20
8	2020	February	Fruit	Apples	11	9
9	2020	February	Fruit	Pears	21	14
10	2020	February	Fruit	Bananas	31	27
11	2020	February	Vegetables	Cucumber	8	6
12	2020	February	Vegetables	Tomatoes	12	10
13	2020	February	Vegetables	Lettuce	20	19

Below, the loaded data is presented in a table visualization.

Year	Month	Category	Type	Sales	Cost
2020	January	Fruit	Apples	12	10
2020	January	Fruit	Pears	21	13
2020	January	Fruit	Bananas	29	26
2020	January	Vegetables	Cucumber	9	6
2020	January	Vegetables	Tomatoes	13	11
2020	January	Vegetables	Lettuce	22	20
2020	February	Fruit	Apples	11	9
2020	February	Fruit	Pears	21	14
2020	February	Fruit	Bananas	31	27
2020	February	Vegetables	Cucumber	8	6
2020	February	Vegetables	Tomatoes	12	10
2020	February	Vegetables	Lettuce	20	19

## Changing the order of the columns in a table

You can change the order of the columns in a table visualization.

- Click a column header, and drag the column horizontally to the desired position.



You can place a column furthest to the left or furthest to the right in the table by clicking its header and selecting **Move first** or **Move last** in the pop-up menu.



You can also go to the **Columns** page in the **Properties dialog** and click a column in the **Selected columns** list and then click **Move up** or **Move down** to change the order of the columns.

## Hiding a column in a table

You can hide columns that do not need to be visible in the table visualization.

### Procedure

- In the table visualization, click the header of the column you want to hide.
- In the opened menu, select **Hide column**.



To display columns that have been hidden, see [Adding columns to a table](#) on page 283.

## Adding columns to a table

You can add columns to your visualization from the underlying data table. This is also how you show columns again that have previously been hidden.

### Procedure

1. Right-click the table visualization, and select **Properties** in the opened menu.  
The Properties popover is shown.
2. Click **Columns**.  
The **Columns** section is shown.
3. Click **Select columns**.  
The Select columns dialog is shown.
4. In the **Available columns** list, select the columns you want to show in the table.
5. Click **Add**.  
The columns are added furthest to the right in the table.



You can click a column in the **Selected columns** list and use the **Move up** and **Move down** buttons to change the display order of the columns.

6. Click **Close**.

## Changing height and width in tables

You can change row height and column width in tables and cross tables to better suit your content.

### Procedure

1. Place the cursor between two rows or two columns in the table, where the cursor changes appearance.
2. Adjust the row height or column width by clicking and dragging the red line that appears.

February	March
34	28
96	100
25	25



Press **Ctrl** simultaneously to adjust all the columns to the same width. The height of header row is adjusted separately, while the data rows always get the same height.

3. Release the cursor at the desired height or width.



To optimize the use of screen estate with all content in the columns kept visible, you can double-click the column divider to adapt the column width to the content. Alternatively, click the column header, and in the opened pop-up menu, select **Size to fit**.

You can also define an exact column width in pixels, by clicking the header and specifying a **Column width** in the pop-up menu.





In the installed client, you can optimize the use of screen estate for the entire table. Right-click the visualization, and select **Resize Columns to Fit**. This will make the column widths as small as possible. That is, every column adjusts automatically to the smallest size possible while keeping all content visible.

## Sorting a table

In a table visualization you can sort the rows by the values in a data column. Depending on the type of data, an alphabetical, numerical, or chronological sorting of the column values is applied. You can also make a subsequent sorting of the rows by the data in another column.

The order can be ascending or descending. In an ascending order, text values are sorted from A to Z, numerical values from lower to higher, and dates from older to newer, and in the descending order, the values are sorted the other way around.

### Procedure

1. In the table, click the header of the column containing the values you want to sort by.
2. In the opened menu, beneath **Sorting**, click  to sort from lower to higher values, or click  to sort the from higher to lower values.  
The rows are sorted by the column values, and an arrow in the column header indicates in which order.



You can click **None** in the menu to return to the default order.



You can click the header while pressing Shift to switch between the sorting modes.

3. If you want to sort by the values in more than one column, click the header of the next column while pressing Shift + Ctrl.  
Within each group of equal values from the first sorting, the rows are sorted by the column values in this second column in an ascending order.



You can repeat this step to sort the rows by the values in more columns.




If you press Shift + Ctrl once again while clicking an already sorted column, the sort order changes to descending.



### Example

The table below lists male and female customers of different ages, and located at two different places.

City	Customer ID	Gender	Age
London	ID1	Female	55
Manchester	ID2	Male	21
London	ID3	Female	68
Manchester	ID4	Male	71
London	ID5	Male	42
Manchester	ID6	Female	35
London	ID7	Male	77
Manchester	ID8	Female	33
London	ID9	Female	40
Manchester	ID10	Male	32
Manchester	ID11	Female	19

To get a more structured view of the table, the City header below has been clicked, followed by  in the opened menu, to gather London and Manchester customers, respectively.

City 	Customer ID	Gender	Age
London	ID1	Female	55
London	ID3	Female	68
London	ID5	Male	42
London	ID7	Male	77
London	ID9	Female	40
Manchester	ID2	Male	21
Manchester	ID4	Male	71
Manchester	ID6	Female	35
Manchester	ID8	Female	33
Manchester	ID10	Male	32
Manchester	ID11	Female	19

Also, within each city, you may want to list male and female customers separately. Simply click the Gender header while pressing Shift + Ctrl. To list Male first, that is, to get a descending order, press the Gender header one more time while pressing Shift + Ctrl.

City ▲	Customer ID	Gender ▼	Age
London	ID5	Male	42
London	ID7	Male	77
London	ID1	Female	55
London	ID3	Female	68
London	ID9	Female	40
Manchester	ID2	Male	21
Manchester	ID4	Male	71
Manchester	ID10	Male	32
Manchester	ID6	Female	35
Manchester	ID8	Female	33
Manchester	ID11	Female	19

If you click the Age header while pressing Shift + Ctrl , the ages will also get sorted within each sorted group.

City ▲	Customer ID	Gender ▼	Age ▲
London	ID5	Male	42
London	ID7	Male	77
London	ID9	Female	40
London	ID1	Female	55
London	ID3	Female	68
Manchester	ID2	Male	21
Manchester	ID10	Male	32
Manchester	ID4	Male	71
Manchester	ID11	Female	19
Manchester	ID8	Female	33
Manchester	ID6	Female	35

## Specifying the header orientation and header text alignment

A column header can be displayed horizontally or vertically in the table visualization. By changing to a vertical header for narrow columns, the screen estate can be used more efficiently. You can also specify the horizontal and vertical alignment of the text within the header.

An example of a table visualization with horizontal and vertical headers is shown below.

Year	Month	Category	Type	Sales	Cost
2018	January	Fruit	Apples	13	11
2018	January	Fruit	Pears	22	14
2018	January	Fruit	Bananas	30	28
2018	January	Vegetables	Cucumber	10	7
2018	January	Vegetables	Tomatoes	14	12
2018	January	Vegetables	Lettuce	23	20
2018	February	Fruit	Apples	12	10
2018	February	Fruit	Pears	22	14
2018	February	Fruit	Bananas	32	29
2018	February	Vegetables	Cucumber	9	11
2018	February	Vegetables	Tomatoes	13	14
2018	February	Vegetables	Lettuce	21	21
2018	March	Fruit	Apples	10	12

By changing to a vertical header orientation for narrow columns, the screen estate can be used more efficiently. See also [Changing height and width in tables](#) on page 283.

#### Procedure

1. Click the column header you want to adjust the orientation or text alignment for.
2. In the **Header** section of the popover, select the **Style this header individually** check box, if you want only this column header to be adjusted. If the check box is not selected, all column headers are adjusted.
3. Use the buttons in the **Header** section to specify the horizontal and the vertical alignment of the header text, and the header orientation.

#### Result

The orientation of the header and the alignment of the header text for the column or columns in question are adjusted accordingly.

## Specifying the text alignment for table cells

You can specify the horizontal alignment of the text for all cells in the table, or for the cells in an individual column.

#### Procedure

1. Click the header of a column for which the text alignment in the table cells should be changed.
2. In the **Values** section of the popover, select the **Style this column individually** check box, if you want the table cells in this column only to be adjusted. If the check box is not selected, the text alignment is adjusted in all table cells.
3. Click the tick mark, **Left**, **Center**, or **Right** button to specify the horizontal alignment.  
The tick mark represents the default alignment, which displays numerical values to the right, and text values to the left.



## Result

The horizontal alignment of the table cell texts in the column or columns in question is adjusted accordingly.

## Specifying text color, background color, and font style in a table

You can specify the text color and background color, as well as the font style for an individual header, all headers, an individual column, or all table cells.

### Procedure

1. Click the header of a column for which you want to change the text style. A popover is opened. It contains styling options categorized in different sections for **Header** and **Values**.
2. Hover the mouse pointer over the **Header** and **Values** sections. The parts of the table that will be affected by the selections you make, will be highlighted.
3. Make the text styling selections of your choice, for example text color  or background color , and then close the popover by clicking outside the popover.



If you want the text styling settings to be applied to one single header or column, select **Style this header individually** or **Style this column individually** respectively.

## Showing text as hyperlinks

You can show the contents of a column as clickable links.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

### Procedure

1. Right-click the table visualization and select **Properties**.
2. Select the **Columns** tab.
3. Select the column you want to contain links.
4. Select **Link from the Renderer** drop down list.
5. Click **Settings**.
6. Specify the settings in the [Renderer settings](#) on page 566 and click OK.

## Showing images in tables

It is possible to show images in the cells of a table visualization. A Spotfire analysis or information link can contain a data table with images that will be shown in the column in the table visualization.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

Continent	Country	City	Sales ▲
Asia		Tokyo	288
Europe		Frankfurt	414
Africa		Johannesburg	667
Asia		Yokohama	993
Asia		Calcutta	1001
Europe		Berlin	1010
Africa		Cape Town	1200
Asia		Bangalore	1221

If the column values are in the form of a URL you can also retrieve images from another location. Use **Image from URL** as the selected Renderer to show linked images.

Another way of including pictures in the table visualization is if the column is a virtual column, collecting the images from a remote data source. For more information on virtual columns, see the [Virtual columns](#) page.

You can change the renderer for a specific column in a table visualization using the following steps:

### Procedure

1. Right-click the table visualization, and select **Properties**.
2. Select the **Columns** tab.
3. Select the column containing the images.
4. Select **Image** from the **Renderer** drop down list.

It is also possible to change the default renderer for different content types from the **Tools > Options** menu. This will affect all future table visualizations, and can be useful if you know that you always want a certain column with a specific content type to be rendered the same way. For more information, see the [Renderer settings](#) on page 566 dialog.



The height of the rows in the table visualization might not be suited for showing images by default. See [Changing height and width in tables](#) on page 283 to change the row height manually.

### Copying the image

It is possible to copy the images from the table. To do this, right-click the cell with the image and select **Copy Cell > Image**.



If the table contains images, these will not be exported if the data from the table visualization is exported to a text file. To preserve the images in the exported file, select **File > Export > Visualization to Microsoft Excel**.



Remember that your internet options settings affect the table visualizations and the text area, so if you have trouble viewing images, make sure that Show pictures is enabled in your web browser.

## Virtual columns

It is possible to include data in a table visualization (or in labels and tooltips of some other visualizations) that is not part of the original data table.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

A virtual column connects to a remote data source and collects data from there. This can be useful if, for example, a large quantity of data is stored in a database and the Spotfire analysis file would be too large if the data was included. The virtual columns can contain plain text but a more common use might be if the collected data were to consist of images or other large files.



Virtual columns are not available by default. A virtual column type needs to be registered programmatically.



Virtual columns are not supported for tables using in-db data sources.

### Differences between virtual columns and regular columns

In the context of the table visualization, there are no major differences except that you cannot sort by the virtual column. However, the virtual column is only part of the table visualization and not part of the data table. This means that:

- No filter will be created for the virtual column.
- No export of the underlying data table will include the virtual column. To export the data, the visualization, and not the data table, must be chosen. Copy will wait until the value has been collected in the virtual column and then copy the value.
- Settings, such as changing the column's name, are not managed from **Data > Column properties**, but from the Virtual columns section in the table visualization properties.
- The virtual column will not be available for selection on axes in the other visualization types.

The virtual column might take some time to load, especially if the column contains images. However, the column will not be loaded all at once. It is possible to start working with the column and the contents will be loaded as you scroll down in the table. The remote data source from where the virtual column is retrieved may require you to enter a username and password.

## Cross table

A cross table is a two-way table consisting of columns and rows. It is also known as a pivot table or a multi-dimensional table.

It is a grid of cells defined by a **Row axis** and a **Column axis**. In some places they are called **Vertical axis** and **Horizontal axis**). On these axes, data columns are specified to organize the data. The actual contents of the table cells are aggregated values, for example, sums or averages of the data, defined by the intersection of the two axes. What data you want to aggregate is selected in another axis, the **Cell values** axis.

Its greatest strength is its ability to structure, summarize and display large amounts of data. You can also use the cross table to determine whether there is a relation between the row variable and the column variable.

	Category		
	Fruit	Spices	Vegetables
Region			
Midwest	344715,22	222585,777	209814,53
Northeast	319687,31	219242,056	310853,64
South	591414,24	394640,1655	492619,1
West	517839,34	361300,3625	324124,3
Sum(Sales)			

1. Row axis (or Vertical axis)
2. Column axis (or Horizontal axis)
3. Cell values

You can change the appearance of the cross table. For example, headers can be displayed horizontally or vertically, and the horizontal as well as the vertical alignment of the header texts can be specified.

### Example

Below is a cross table showing the sum of sales for different types of fruits and vegetables.

Category	Type	2020	2021	2022	Grand total
Fruit	Apples	150	153	162	465
	Bananas	332	336	344	1012
	Pears	267	266	279	812
Vegetables	Cucumber	140	141	152	433
	Lettuce	246	245	258	749
	Tomatoes	156	161	168	485
Grand total		1291	1302	1363	3956

The same data displayed in a regular table takes up far more space and it is much more difficult to get an overview:

Year	Month	Category	Type	Sales	Cost
2020	January	Fruit	Apples	12	10
2020	January	Fruit	Pears	21	13
2020	January	Fruit	Bananas	29	26
2020	January	Vegetables	Cucumber	9	6
2020	January	Vegetables	Tomatoes	13	11
2020	January	Vegetables	Lettuce	22	20
2020	February	Fruit	Apples	11	9
2020	February	Fruit	Pears	21	14
2020	February	Fruit	Bananas	31	27
2020	February	Vegetables	Cucumber	8	6
2020	February	Vegetables	Tomatoes	12	10
2020	February	Vegetables	Lettuce	20	19
2020	March	Fruit	Apples	9	10
2020	March	Fruit	Pears	19	13
2020	March	Fruit	Bananas	32	27
2020	March	Vegetables	Cucumber	8	6
2020	March	Vegetables	Tomatoes	11	11
2020	March	Vegetables	Lettuce	21	19
2020	April	Fruit	Apples	9	10
2020	April	Fruit	Pears	18	13
2020	April	Fruit	Bananas	32	27
2020	April	Vegetables	Cucumber	10	7
2020	April	Vegetables	Tomatoes	12	11

You can display grand totals for columns and rows (the example above shows grand totals for columns as well as rows). See [Displaying totals](#) on page 294 to learn more about totals.


## Creating a cross table

Cross tables are used to summarize large amounts of data, and then present the result in a structured table format.

### Prerequisites

See [Cross table](#) on page 290 for a description of the concept.

### Procedure

1. On the authoring bar, click  to open the **Visualization types** flyout.
2. Drag the **Cross table** visualization type to the wanted position on the analysis page. A suggestion of a cross table is presented.
3. On the **Row axis** (sometimes called **Vertical axis**), [select one or more columns](#) containing the categories that you want to use for organizing the rows in the cross table.
4. On the **Column axis** (sometimes called **Horizontal axis**), select one or more columns containing the categories that you want to use for organizing the columns in the cross table.



When you select more than one column on the Row axis or Column axis, you are setting up a [hierarchy](#), which is a very powerful feature in a cross table.

5. On the **Cell values** axis, select the column whose values you want to aggregate.
6. [Select which type of aggregated value](#) you want to use.



You can also set up your cross table by dragging a column of interest from the **Data in analysis flyout** and dropping it on one of the three drop targets shown.







## Displaying totals

You can display grand totals for columns as well as rows in a cross table. If the vertical axis is hierarchically structured, it is possible to display also column subtotals for different levels in the hierarchy.

		Gender			
Store Location » Department	Store Location	Department	Female	Male	Grand total
	Boston	1	938	1252	2190
		2	3929	1092	5021
		3	10248	2247	12495
		4	16524	13682	30206
		5	46046	27318	73364
		6	91545	54472	146017
		Subtotal	169230	100063	269293
	Seattle	1	0	153	153
		2	2850	1521	4371
		3	3771	2974	6745
		4	13246	11061	24307
		5	43342	15820	59162
		6	64157	30417	94574
		Subtotal	127366	61946	189312
	Grand total		296596	162009	458605
	Sum(Sales)				

Totals are calculated using the same aggregation or expression as is used for calculating the cell values. The calculation is not based on the aggregated values shown in the cross table cells; it is by default based on the underlying data table rows. For example, if 'Average' is used as aggregation method for the cell values, the grand total average of the column is not the average of the cell values; it is the average of all the underlying data rows. This is exemplified below, where a simple data table is visualized by a cross table:

Region	Number
East	200
East	300
West	1
West	9
East	100

(None)	
Region	Avg(Number)
East	200.00
West	5.00
Grand total	122.00
Avg(Number)	

The expression used for calculating the cell values is average (Avg). The Grand total is calculated on the underlying rows  $((200+300+1+9+100)/5=122)$ , and not on the cell values.



There is, however, an option to [base a total on the already aggregated cell values](#) in the cross table. If you use this option, the total will be calculated as the sum of the cell values, no matter the expression used for calculating the cell values. See [Displaying grand totals and subtotals as the sum of the cell values](#).

### Displaying grand totals for columns or rows in a cross table

You can show a grand total for each column or each row in a cross table. A grand total is by default calculated using the same expression that is used for calculating the cell values. For example, a grand total can indicate a 'grand sum' or a 'grand average'.

In the cross tables below, the aggregation method 'Sum' is used to aggregate the 'Sales' values, and accordingly, the Grand totals are also calculated as a 'Sum'.

In this example, grand totals for each column are shown.

		Year » Month				
Country » City	Country	City	2021		2022	
			January	February	January	February
	France	Lyon	22	35	15	29
		Marseille	13	41	21	40
		Paris	9	54	10	60
	Germany	Berlin	12	45	15	60
		Hamburg	21	23	24	44
		Munich	29	38	30	33
Grand total		106	236	115	266	
Sum(Sales)						

In this example, grand totals for each row are shown.

		Year » Month					
Country » City	Country	City	2021		2022		Grand total
			January	February	January	February	
	Country » City	France	Lyon	22	35	15	
Marseille			13	41	21	40	
Paris			9	54	10	60	
Germany		Berlin	12	45	15	60	
		Hamburg	21	23	24	44	
		Munich	29	38	30	33	
Sum(Sales)							

Totals are not based on the aggregated values shown in the cross table cells; they are by default based on the underlying data table rows. For example, if 'Average' is used as aggregation method, the Grand total average of a column takes into account all the data rows on which the cell values are based, and not only the averages shown in the cross table cells. See [Displaying totals](#) on page 294.



There is, however, an [option to base a total on the already aggregated cell values](#) in the cross table. If you use this option, the total will be calculated as the sum of the cell values, no matter the expression used for the calculations of the cell values.

### Procedure

1. Right-click the cross table, and select **Properties** in the opened menu.
2. Click **Totals**.
3. Under **Display totals**, select the **Grand total for columns** or the **Grand total for rows** check box.

### Result

The selected grand total is displayed in the cross table, column grand totals at the bottom of the cross table, and row grand totals furthest to the right.



In the installed Spotfire client, you can also right-click the visualization and select **Grand total for columns** or **Grand total for rows** in the opened menu.

## Showing subtotals in cross table columns

In a cross table, you can display column subtotals, that is partial summarizations.

To show subtotals within the cross table columns, the vertical axis (or Column axis) must be structured hierarchically with at least two levels.

In the hierarchy below, subtotals for the month columns are displayed on the 'Country' level. The aggregation 'Sum' is used to aggregate the 'Sales' values, and accordingly, the subtotals are also calculated using the 'Sum' aggregation.

		Year » Month				
Country » City	Country	City	2021		2022	
			January	February	January	February
		France	Lyon	22	35	15
Marseille			13	41	21	40
Paris			9	54	10	60
Subtotal			44	130	46	129
Germany		Berlin	12	45	15	60
		Hamburg	21	23	24	44
		Munich	29	38	30	33
		Subtotal	62	106	69	137
Sum(Sales)						

If the hierarchy on the vertical axis has more than two levels, you can specify on which levels subtotals should be displayed.

Subtotals are calculated using the same aggregation or expression that is used for the calculation of the cell values. The subtotals are not based on the aggregated values shown in the cross table cells; they are by default based on the underlying data table rows. For example, if 'Average' is used as the aggregation method, the subtotal average of a column takes into account all the data rows on which the cell values are based, and not only the averages shown in the cross table cells. See [Displaying totals](#) on page 294.



There is, however, an [option to base a total on the already aggregated cell values](#) in the cross table. If you use this option, the total will be calculated as the sum of the cell values, no matter the expression used for the calculations of the cell values.

### Procedure

1. Right-click the cross table, and select **Properties** in the opened menu.
2. Click **Totals**.
3. Under **Display totals**, select the levels in the hierarchy for which subtotals should be shown. (In the hierarchy above, there is only one possible level.)
4. Specify where in the cross table column the subtotals should be displayed, **Before values** or **After values**.

### Result

The selected subtotals are displayed in the cross table.



In the installed Spotfire client, you can also right-click the visualization and select **Subtotals for Columns**. A submenu lists the columns and/or hierarchies levels you can show subtotals for on the vertical axis (or Column axis).

## Displaying grand totals and subtotals as sums of the cell values

By default, grand totals and subtotals in a cross table are calculated using the same aggregation or expression that is used for calculation of the cell values, and the calculations are based on the underlying data table rows. However, it is possible to base the totals calculations on the aggregated cell

values. Then totals will be calculated as the sum of the cell values, no matter the expression used for the cell values.

### Procedure

1. Right-click the cross table, and select **Properties** in the opened menu.
2. Select **Totals**.
3. Under **Display totals**, select the totals to be visible in the cross table.
4. If any subtotals are displayed, select their position; **Before values** or **After values**.
5. **Settings of calculations** lists the expressions used for calculating the cell values for the different expressions on the Cell values axis, and what data the calculations are based on. Click the expression, whose totals you want to calculate as a sum of the cell values.
6. Select **As sum of cell values** under **Calculate totals**.

### Result

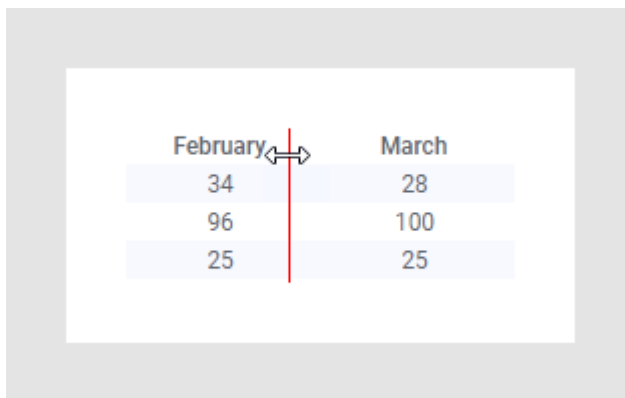
Any totals for the expressions selected in the last step are calculated as sums. It is also indicated in the list that they are calculated as **Sum of cell values**.

## Changing height and width in tables

You can change row height and column width in tables and cross tables to better suit your content.

### Procedure

1. Place the cursor between two rows or two columns in the table, where the cursor changes appearance.
2. Adjust the row height or column width by clicking and dragging the red line that appears.



February	March
34	28
96	100
25	25



Press **Ctrl** simultaneously to adjust all the columns to the same width. The height of header row is adjusted separately, while the data rows always get the same height.

3. Release the cursor at the desired height or width.



To optimize the use of screen estate with all content in the columns kept visible, you can double-click the column divider to adapt the column width to the content. Alternatively, click the column header, and in the opened pop-up menu, select **Size to fit**.

You can also define an exact column width in pixels, by clicking the header and specifying a **Column width** in the pop-up menu.



In the installed client, you can optimize the use of screen estate for the entire table. Right-click the visualization, and select **Resize Columns to Fit**. This will make the column widths as small as possible. That is, every column adjusts automatically to the smallest size possible while keeping all content visible.

## Sorting values in a cross table



You can sort the aggregated values in the cross table columns as well as in the cross table rows.

You can sort the values in ascending or descending order. In hierarchically structured cross tables, the sorting of values is made within each of the smallest subcategories.



You can also sort the values in a cross table column globally, that is, a sorting that ignores any specified hierarchy.

### Procedure

1. Click the header of the cross table column or row that contains the values you want to sort.
2. In the opened menu, beneath **Sorting**, click  to sort from lower to higher values, or click  to sort the from higher to lower values. When the values are sorted, an arrow in the header indicates in which order.



If the cross table is structured hierarchically, the values are sorted within each of the smallest subcategories. See the example at the bottom.



You can click **None** to return to the default order.





You can press Shift + click the header to switch the sort order.


## Examples

The cross table below visualizes data hierarchically on the horizontal axis as well as on the vertical axis. It shows quarterly sales figures for 2021 and 2022, split by location and gender.

Store Location	Gender	2021			2022		
		Q1	Q2	Q3	Q1	Q2	Q3
Boston	Female	43761	33496	13434	7859	19669	36777
	Male	17713	16836	4473	2901	4842	17881
Los Angeles	Female	26687	22424	10294	9408	21000	42611
	Male	12777	27554	3732	4770	11787	30926
New York	Female	26460	31884	3867	14290	16751	58726
	Male	18281	22659	5864	14823	2368	20775
Seattle	Female	31888	29918	8858	10669	7127	62890
	Male	12465	16364	1304	3116	8772	17292

If you click a column header, for example Q3 in 2021, and select  in the menu, the values are sorted from lower to higher values within the last subcategories of the hierarchy. That is, the male and female sales figures are sorted in ascending order within each store location.

Store Location	Gender	2021			2022		
		Q1	Q2	Q3 	Q1	Q2	Q3
Boston	Male	17713	16836	4473	2901	4842	17881
	Female	43761	33496	13434	7859	19669	36777
Los Angeles	Male	12777	27554	3732	4770	11787	30926
	Female	26687	22424	10294	9408	21000	42611
New York	Female	26460	31884	3867	14290	16751	58726
	Male	18281	22659	5864	14823	2368	20775
Seattle	Male	12465	16364	1304	3116	8772	17292
	Female	31888	29918	8858	10669	7127	62890

If you instead click a row header, for example Male in New York as shown below, and select  in the menu, the values within each of the last subcategories, Q1-Q3 2021 and Q1-Q3 2022, are sorted in an ascending order.



Store Location	Gender	2021			2022		
		Q3	Q1	Q2	Q2	Q1	Q3
Boston	Female	13434	43761	33496	19669	7859	36777
	Male	4473	17713	16836	4842	2901	17881
Los Angeles	Female	10294	26687	22424	21000	9408	42611
	Male	3732	12777	27554	11787	4770	30926
New York	Female	3867	26460	31884	16751	14290	58726
	Male	5864	18281	22659	2368	14823	20775
Seattle	Female	8858	31888	29918	7127	10669	62890
	Male	1304	12465	16364	8772	3116	17292



### Sorting values in cross table ignoring the hierarchy

You can sort the values in a cross table globally, that is, any created hierarchy in the cross table will be ignored when sorting. You can sort the values in ascending or descending order.



Only the values in a cross table column can be sorted globally.

#### Procedure

1. Click the header of the cross table column whose values you want to sort.
2. In the opened menu, beneath **Sorting**, click  to sort from lower to higher values, or click  to sort the from higher to lower values.  
The values in the entire column are sorted, and the hierarchical structure is no longer displayed. An arrow in the header indicates the sort order.



You can click **None** to return to the default order.




You can press Shift + click the header to switch the sort order.

### Example

The cross table below visualizes data hierarchically. It shows quarterly sales figures for 2021 and 2022, split by location and gender.

	Store Location	Gender	2021			2022		
			Q1	Q2	Q3	Q1	Q2	Q3
Store Location » Gender	Boston	Female	43761	33496	13434	7859	19669	36777
		Male	17713	16836	4473	2901	4842	17881
	Los Angeles	Female	26687	22424	10294	9408	21000	42611
		Male	12777	27554	3732	4770	11787	30926
	New York	Female	26460	31884	3867	14290	16751	58726
		Male	18281	22659	5864	14823	2368	20775
	Seattle	Female	31888	29918	8858	10669	7127	62890
		Male	12465	16364	1304	3116	8772	17292

If you click a column header, for example Q3 in 2021, and select  in the menu, the values within the entire column are sorted from lower to higher values. In consequence of this, the hierarchical structure can no longer be displayed.

	Store Location	Gender	2021			2022		
			Q1	Q2	Q3 ▲	Q1	Q2	Q3
Store Location » Gender	Seattle	Male	12465	16364	1304	3116	8772	17292
	Los Angeles	Male	12777	27554	3732	4770	11787	30926
	New York	Female	26460	31884	3867	14290	16751	58726
	Boston	Male	17713	16836	4473	2901	4842	17881
	New York	Male	18281	22659	5864	14823	2368	20775
	Seattle	Female	31888	29918	8858	10669	7127	62890
	Los Angeles	Female	26687	22424	10294	9408	21000	42611
	Boston	Female	43761	33496	13434	7859	19669	36777

### Styling in the cross table

You can change the appearance of the cross table in many different ways.

For example, you can choose to show headers either horizontally or vertically. By using vertical headers for narrow columns, the screen estate can be used more efficiently. You can use text color and background color to highlight specific rows or columns. You can also specify text alignment, as well as bold and italic font style. You can style all the headers on the same hierarchical level, or an individual header. You can style an entire column or row, as well as all the cell values in the cross table.

		2019			2020			Grand total
		January	February	March	January	February	March	
France	Lyon	22	35	21	15	29	22	144
	Marseille	13	41	11	21	40	11	137
	Paris	9	54	8	10	60	8	149
	Subtotal	44	130	40	46	129	41	430
Germany	Berlin	12	45	9	15	60	9	150
	Hamburg	21	23	19	24	44	19	150
	Munich	29	38	32	30	33	35	197
	Subtotal	62	106	60	69	137	63	497
Grand total		106	236	100	115	266	104	927



The text and background coloring that is specified in the styling popover is different from the value-based coloring. Any value-based coloring that is configured in the cross table will override the styling colors. See [Working with colors](#) on page 539 to learn more about value-based coloring.

### Procedure

1. Click a header in the cross table to open a popover.
2. Hover the mouse pointer over the **Header** and **Values** sections. The parts of the cross table that will be affected by the selections you make, will be highlighted.



You can move the popover out of the way if it hides the parts of the cross table that you need to see when you are styling it. Just point the mouse cursor at the top of the popover, then click and drag it to another position.

3. Make the styling selections of your choice, then close the popover by clicking outside the popover.



Which styling settings are available in the popover depends on which header you click. To reach styling settings for the values, open the popover from one of the headers closest to the values. For example Lyon, Hamburg, or March in the example. Alignment settings for the values are available from the headers on the horizontal axis.

4. Read the help topics below with detailed instructions to learn more about specific ways to style the cross table.

### Specifying header orientation and header text alignment for multiple headers

The header texts in cross tables can be displayed horizontally or vertically. You can also specify the horizontal and vertical alignment of the text within the header.

An example of a cross table with different header orientations and text alignments is shown below. Within a header, the horizontal alignment of the text can be set to **Left**, **Center**, or **Right**, and the vertical alignment to **Top**, **Middle**, or **Bottom**.

Country	City	2015			2016		
		January	February	March	January	February	March
France	Lyon	22	35	21	15	29	22
	Marseille	13	41	11	21	40	11
	Paris	9	54	8	10	60	8
Germany	Berlin	12	45	9	15	60	9
	Hamburg	21	23	19	24	44	19
	Munich	29	38	32	30	33	35

By changing to a vertical header orientation for narrow cross table columns, you can use the screen estate more efficiently. See also [Changing height and width in tables](#) on page 283.

#### Procedure

1. Click the header of a cross table column or row.
2. In the **Header** section of the popover, click the buttons to specify the horizontal text alignment, the vertical text alignment, and the header orientation.

#### Result

All headers on the same hierarchical level as the clicked header are adjusted at the same time.



You can also adjust headers individually. For details, see [Specifying header orientation and header text alignment for an individual header](#).

### Specifying header orientation and header text alignment for an individual header

Usually, you specify header orientation and header text alignment for multiple headers in the cross table at the same time. It is possible, though, to make these settings for an individual header.

The header texts in cross tables can be displayed horizontally or vertically. You can also specify the horizontal and vertical alignment of the text within the header.

#### Procedure

1. Click the header you want to adjust.
2. In the **Header** section of the popover, select the **Style this header individually** check box.
3. Click the buttons to specify the horizontal text alignment, the vertical text alignment, and the header orientation for this header.

#### Result

The settings are applied to this header only.



If you select the **Style this header individually** check box, then the alignment and orientation changes made above only affect this individual header, and any adjustments made to other headers are not applied to this header. If you clear the check box, the header will be styled in the same way as other headers. For details, see [Specifying header orientation and header text alignment for multiple headers](#).

## Specifying horizontal text alignment for cross table cells

Usually, you specify the horizontal alignment of the calculated values in the cross table cells for all cross table cells at the same time. It is possible, though, to make this setting for an individual column only.

### Procedure

1. Click the header of a column for which the text alignment in the table cells should be changed.
2. In the **Values** section of the popover, select the **Style this column individually** check box, if you want to adjust the table cells in this column only. If the check box is not selected, the text alignment in all cross table cells is adjusted.
3. Click the tick mark, **Left**, **Center**, or **Right** button to specify the horizontal alignment.  
The tick mark represents the default alignment, which displays numerical values to the right, and text values to the left.



### Result

The horizontal alignment of the table cell texts in the column or columns in question is adjusted accordingly.

## Specifying text color and background color for multiple headers

It is often convenient to change the appearance of all the cross table headers on the same hierarchy level.

### Procedure

1. Click the header of a column or row to open the popover containing the styling settings.
2. In the **Header** section of the popover, click the text color button  to open the color selector and choose a text color. If you want to change the color of the cells rather than the text color, click the background color button  and choose a color from that color selector instead.

### Result

All the headers on the same hierarchical level as the header you clicked will get the new text or background color.





By default, all headers on the same hierarchy level are adjusted. It is also possible to adjust headers individually. See [Specifying text color and background color for an individual header](#) on page 305.

## Specifying text color and background color for an individual header

Sometimes it is useful to change the appearance of a certain cross table header to make it stand out compared to other headers.

### Procedure

1. Click the header you want to adjust individually, to open the popover containing the styling settings.

2. In the **Header** section of the popover, select the **Style this header individually** check box. The highlight in the cross table will change to surround the selected header only.
3. Click the text color button  to open the color selector and choose a text color. If you want to change the color of the header cell rather than the header text, click the background color button  and choose a color from that color selector instead.

### Result

Only the selected header is changed.





When the **Style this header individually** check box is selected, then the styling affects only this individual header, and adjustments made to other headers do not have any effect on this header. If you clear the check box, the header will get the same styling as the other headers.

## Specifying text color and background color for cross table cells

You can specify the text or background color for all cross table cells, or for the cells in an individual column or row.

### Procedure

1. Click the header of a column for which you want to change text color or background color.
2. In the **Values** section of the popover, select the **Style this column individually** check box, if you want to change the color in the selected column only. If the check box is not selected, the color in all cross table cells will be adjusted.
3. Click the text color button  to open the color selector and choose a text color. If you want to change the color of the cells rather than the text, click the background color button  and choose a color from that color selector instead.

### Result

Depending on the selections you made, the color of either the text or the background will be updated for an individual column or all of the cells in the cross table.



To change the color on an individual row instead of an individual column, just select the row header of interest instead, and make sure you select the check box **Style this row individually**.



## Specifying text color, background color, and font style for subtotals and grand totals

When the cross table shows subtotals or grand totals, it can be useful to change the color and font style to make them stand out compared to other rows and columns.

In this example, both subtotals and grand totals have been added to the cross table. The subtotal headers and values are displayed with a different text color, and the grand total headers and values have a different background color.

Country	City	2019			2020			Grand total
		January	February	March	January	February	March	
France	Lyon	22	35	21	15	29	22	144
	Marseille	13	41	11	21	40	11	137
	Paris	9	54	8	10	60	8	149
	Subtotal	44	130	40	46	129	41	430
Germany	Berlin	12	45	9	15	60	9	150
	Hamburg	21	23	19	24	44	19	150
	Munich	29	38	32	30	33	35	197
	Subtotal	62	106	60	69	137	63	497
Grand total		106	236	100	115	266	104	927

### Procedure

1. Click the subtotal header for which you want to change the text style.
2. Hover the mouse pointer over the **Subtotal header** and **Subtotal values** sections in the popover that opened.  
The subtotal headers and the subtotal values will be highlighted in the cross table as you move the mouse pointer between the sections.
3. Click the text color icon  in the **Subtotal header** section to open the color selector and choose a text color. If you want to change the color of the cells rather than the text, click the background color button  and choose a color from that color selector instead.  
Depending on the selection you made, either the text color or the background color of the subtotal headers is changed to the new color.
4. To get the same color on the entire subtotal row, like the example above, just select the same color in the **Subtotal values** section.
5. To style the text or background color for grand totals, click one of the grand total headers instead.  
Note that grand total for columns and grand total for rows are styled separately.

### Result

The rows and columns showing subtotals and grand totals are now easier to spot in the cross table.

## Formatting in the cross table

It is possible to format the values in the cross table on any of the axes. You can format the values either by opening the Formatting section of the visualization properties, or, in the installed client, by right-clicking directly in the cross table. The example below illustrates how formatting can be applied to a cross table in the installed client.

### Example

The cross table below shows the sum of sales (Sum(Sales)) for fruits and vegetables for the years 2020, 2021, and 2022. The horizontal axis has the two columns Category and Type. The column Category contains the two axis values Fruit and Vegetables, and the column Type contains the values Apples, Bananas, Pears, Cucumber, Lettuce, and Tomatoes.

Year	Fruit			Vegetables		
	Apples	Bananas	Pears	Cucumber	Lettuce	Tomatoes
2020	150	332	267	140	246	156
2021	153	336	266	141	245	161
2022	162	344	279	152	258	168

To format all the values for 2020 as currency, right-click the row header 2020, choose Formatting and then select Currency from the pop-up menu. The values in that row will be updated instantly with the new setting. To format the row containing the values for 2022, go through the same steps. The resulting cross table is shown below.

Year	Fruit			Vegetables		
	Apples	Bananas	Pears	Cucumber	Lettuce	Tomatoes
2020	\$150.00	\$332.00	\$267.00	\$140.00	\$246.00	\$156.00
2021	153	336	266	141	245	161
2022	\$162.00	\$344.00	\$279.00	\$152.00	\$258.00	\$168.00



To select another currency you must open the properties dialog.

You can go through the same steps again to format the values for 2021. However, if you format the values for the Apples column instead, all the previous formatting settings will be reset, as seen in the cross table below.

Year	Fruit			Vegetables		
	Apples	Bananas	Pears	Cucumber	Lettuce	Tomatoes
2020	\$150.00	332	267	140	246	156
2021	\$153.00	336	266	141	245	161
2022	\$162.00	344	279	152	258	168

Only the values in the column Apples are now formatted as currency, because you can only format values in one direction and at one level at a time. This means that if you format the values for Vegetables, then the values in Apples will be reset, as seen in the cross table below.



Year	Fruit			Vegetables		
	Apples	Bananas	Pears	Cucumber	Lettuce	Tomatoes
2020	150	332	267	\$140.00	\$246.00	\$156.00
2021	153	336	266	\$141.00	\$245.00	\$161.00
2022	162	344	279	\$152.00	\$258.00	\$168.00

You can move a measure from one axis to another without losing the formatting settings you have defined. This means that if you were to move the column Category from the horizontal axis to the vertical axis, all the values for the Vegetables would still be formatted as currency.

If you right-click any of the cells that are not row or column headers in the cross table and select a formatter, all the values in the cross table will get the new formatting settings.

However, if you add another column to the cell values axis it will only be possible to format on cell values. In the cross table below, Sum(Cost) was added to the cell values axis, and the column Type was removed.

Year	Fruit		Vegetables	
	Sum(Sales)	Sum(Cost)	Sum(Sales)	Sum(Cost)
2020	749	563	542	446
2021	755	584	547	476
2022	785	624	578	501

With this configuration, it is no longer possible to format only the values for an individual year, nor the values only for Fruit or Vegetables. If you right-click on any of those header cells in the cross table, the pop-up menu will not provide any formatting options to choose from. However, you can still format the values on Sum(Sales) and Sum(Cost). If you right-click one of the two header cells for Sum(Sales) and set formatting to Currency, the cross table will look like the one below.

Year	Fruit		Vegetables	
	Sum(Sales)	Sum(Cost)	Sum(Sales)	Sum(Cost)
2020	\$749.00	563	\$542.00	446
2021	\$755.00	584	\$547.00	476
2022	\$785.00	624	\$578.00	501

## Marking values in cross tables

When you want to further explore the data summarized in the cross table cells, it is possible to mark all cell values in a column or in a row simultaneously.

For information on different ways of exploring data that is marked, see [Marking](#).

If the cross table is hierarchically structured, you can mark all values in the columns or rows under a higher level header in the hierarchy instantly.

### Procedure

1. Click the header whose values you want to mark.

You can click a header on the horizontal axis, or a header on the vertical axis.

2. In the opened menu, select **Mark values**.  
The cell values subordinate to the header are marked.



Alternatively, you can press Alt + click the header to mark the values.



Clicking a header on the lowest level in a hierarchy will mark all cell values in the column or row. Clicking a header on a higher level in the hierarchy will mark all values subordinate to the header.

## Examples

In the cross table below, **Mark values** has been clicked for the header 'February'. Every value in the 'February' column is marked.

		Year » Month						
Category » Type	Category	Type	2020			2021		
			January	February	March	January	February	March
	Fruit	Apples	12	11	9	13	11	9
		Bananas	29	31	32	29	33	35
		Pears	21	21	19	21	21	19
	Vegetables	Cucumber	9	8	8	9	8	8
Lettuce		22	20	21	21	22	22	
Tomatoes		13	12	11	13	12	11	
Sum(Sales)								

In the next cross table, **Mark values** has been clicked for the header 'Vegetables'. Because 'Vegetables' is on a higher level in the hierarchy, the values in all its sub-headers are marked.

		Year » Month						
Category » Type	Category	Type	2020			2021		
			January	February	March	January	February	March
	Fruit	Apples	12	11	9	13	11	9
		Bananas	29	31	32	29	33	35
		Pears	21	21	19	21	21	19
	Vegetables	Cucumber	9	8	8	9	8	8
		Lettuce	22	20	21	21	22	22
Tomatoes		13	12	11	13	12	11	
Sum(Sales)								

## Images in the cross table

Cross tables can be configured to show images in axis labels.

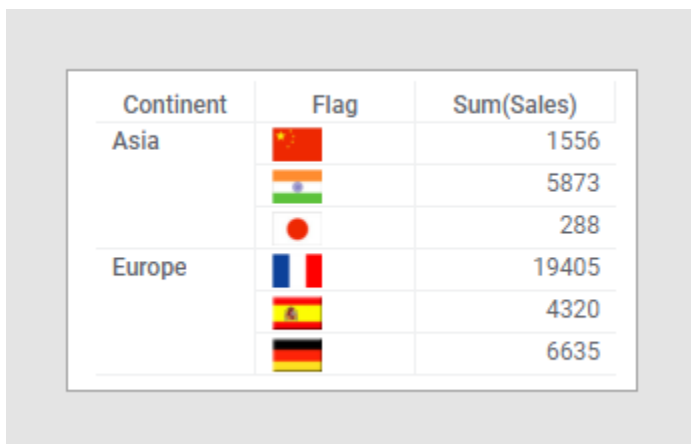








Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

You can show images using one of the following options:

- Images are fetched via an URL and built up by the content of the selected expression/column. The actual image is then stored in a file on the network, or on the Internet, and is rendered using the Image from URL option in the Label rendering section.
- Images are stored in BLOB columns in an in-memory data table and rendered directly as images.
- A virtual column producer is available which can transform the value from the chosen expression/column into a desired output. For example, a column containing the IDs of molecules can be used

on the axis and the corresponding Molfiles can be fetched from a database and rendered as images showing chemical structures in Spotfire.



Continent	Flag	Sum(Sales)
Asia		1556
		5873
		288
Europe		19405
		4320
		6635

### Setting up a cross table with images from a binary column or from a URL

1. In the visualization properties for the visualization of interest, locate the **Row axis** and **Column axis** sections.



The visualization properties look slightly different depending on the client you use and from where you open the properties, but the settings work the same. The Row axis and Column axis are sometimes called Vertical axis and Horizontal axis.

2. On the **Row axis** or **Column axis** section, in the column selector, choose the column containing binary data to show images based on binary data.

To show images from the web or a local network using a URL, choose the column containing the identifier of the image.

3. Click the axis section to expand it and to see the **Label rendering** settings.
4. If you have selected more than one column on the axis, click the column of interest.
5. To show images from a binary column, change **Show as** to **Image**.

To show images from a URL, change **Show as** to **Image from URL**.

6. If you chose **Image from URL**, you must configure the URL. Click **Settings...** or the settings icon next to the drop-down list. **Enter the URL where {\$} represents each cell value**, so that the resulting URL points to your image when the values from the identifier column replaces the {\$}. For example, if the images are stored on a network drive which is common for everybody in your company and the identifier column contains the names of the images, then the URL to enter might be Q:\Images \{\$}.
7. Click **OK**.

## Graphical table

A graphical table is a summarizing visualization designed to provide a lot of information at one glance.



Graphical tables must be authored in the installed client.

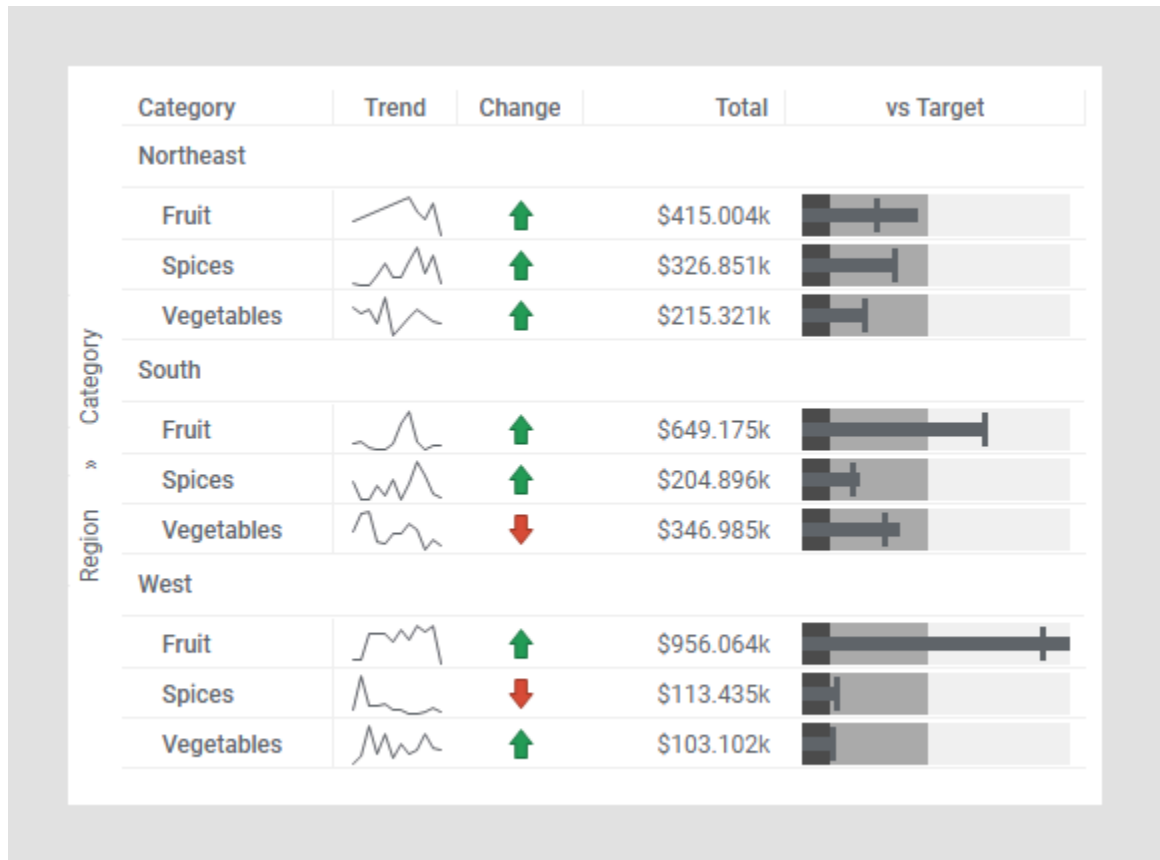
A graphical table can be set up to show columns with items such as

- **sparklines** that are line charts in miniature
- **calculated values** that are derived from aggregations or expressions

- conditional [icons](#) that indicate certain conditions
- [bullet graphs](#) that compare two values symbolized by a horizontal bar and a vertical line on a colored background.

The items are dynamic, which means they respond instantly to filtering and other property changes.

In the graphical table below, the dynamic items show sales performance for different categories within each region.



You can add any number of dynamic items to a graphical table. Each dynamic item column uses its own axis expression and it can also be filtered and limited by markings separately. This way, you can show both the total values for some calculated value and the currently filtered values simultaneously.

When a hierarchical structure is set up, the graphical table is grouped into sections and sorting can be performed within each section by clicking a column header.

## Creating a graphical table

A graphical table is a summarizing visualization designed to provide a lot of information at one glance.



Graphical tables must be authored in the installed client.

### Procedure

1. On the authoring bar, click to open the **Visualization types** flyout.
2. Drag the **Graphical table** visualization to the desired position on the analysis page.



You can change which item type to be used as default under **Tools > Options > Graphical table**.

- Adjust the graphical table to display the items of your choice.


## Adding dynamic items to a graphical table

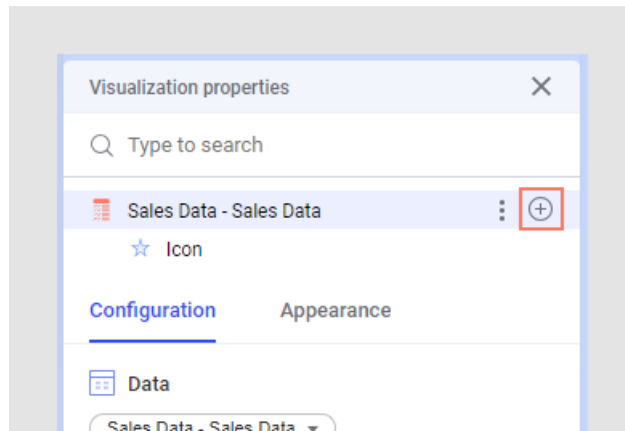
A graphical table can be set up to show columns with dynamic items such as sparklines, calculated values, conditional icons, or bullet graphs.



In the web client, you cannot create a new graphical table visualization. You can only configure an existing one that has been created using the installed client.

### Procedure


- Create or click to activate a graphical table.
- In the visualization properties panel, click  next to the title of the graphical table.




- From the pop-up menu, select the type of dynamic item you want to add in a new column: **Sparkline**, **Calculated value**, **Icon**, or **Bullet graph**.




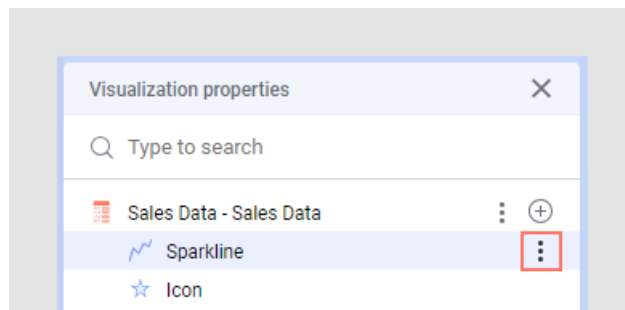
In the installed client, you can also add a new column by right-clicking the graphical table and selecting the type of dynamic item you want to add from the **Add** submenu.

- Below the visualization title, make sure the dynamic item you want to adjust is selected. The visualization properties panel now only shows settings that are applicable to the selected item. You can also click  **Add** to add and configure more properties.
- Make the necessary changes to the settings. See below for specific settings for each of the dynamic items.

Option	Description
<b>Sparkline</b>	<p>In the <b>X-axis</b> and the <b>Y-axis</b> sections, select the columns, hierarchies, or expressions to define the axes.</p> <p> In the installed client, you can also right-click the sparkline column and select <b>Sparkline settings &gt; Axes</b>.</p>
<b>Calculated value</b>	<p>In the installed client, right-click the new column, and select <b>Calculated value settings</b> from the pop-up menu. On the <b>Values</b> page, select a column, hierarchy, or expression in the <b>Calculate values using</b> field.</p>
<b>Icon</b>	<p>In the installed client, right-click the new column, and select <b>Icon settings</b> from the pop-up menu. On the <b>Icons</b> page, select a column, hierarchy, or expression in the <b>Calculate icons using</b> field.</p>
<b>Bullet graph</b>	<p>In the installed client, right-click the new column, and select <b>Bullet graph settings</b> from the pop-up menu.</p>

Option	Description
	<p>On the <b>Bullet graphs</b> page, select a column or expression in the <b>Calculate values using</b> field. This is the horizontal bar in the bullet graph.</p> <p>Select a column or expression in the <b>Calculate comparative values using</b> field. If you do not want to use a comparative value, select <b>Remove</b> in the column selector. This is the vertical line in the bullet graph, sometimes also referred to as the target value.</p> <p>If desired, change the <b>Color</b> of the bar and/or vertical line. You can also <a href="#">add color ranges</a>.</p>

- To change the name of the dynamic item to something that describes the dynamic item column content, click  and select **Rename column**. You can also [hide the column name or the entire header row](#) of the graphical table.



In the installed client, you can also configure these settings under **General** and **Appearance**.

- If you want, make other changes to the default settings, such as changing the filtering scheme under **Data**, adding an action, [displaying starting point and end point Y-axis values](#) (sparkline), setting up a rule (calculated value), or adding color ranges (bullet graph).

### Changing the column order

Click a column header and drag that column to the desired position. Note that the row header column cannot be moved.

In the installed client, you can also right-click a column header and select **Move first** or **Move last** from the pop-up menu.

### Removing a column from a graphical table


See [Removing a dynamic item from a graphical table](#) on page 315 for how to remove a column.

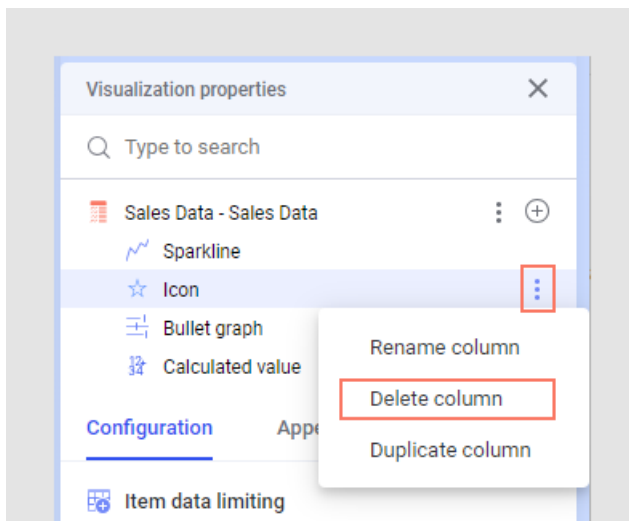
## Removing a dynamic item from a graphical table

You can remove columns with dynamic items from a graphical table.



Make sure the graphical table you are working with is selected on the analysis page.

In the visualization properties panel below the visualization name, click  on the column you want to remove, and select **Delete column**.



In the installed client, you can also right-click the column header of the dynamic item column you want to remove and select **Remove** from the pop-up menu.


## Changing the column header for a dynamic item

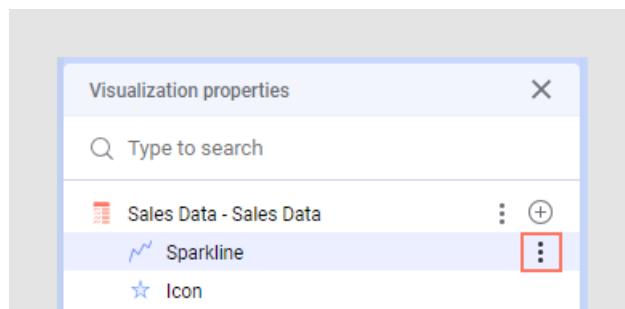
You can change the name of a column containing dynamic items.

### Prerequisites

Make sure the graphical table you are working with is selected on the analysis page.

### Procedure

- In the visualization properties panel, locate the column that you want to rename.  
On the column, click  and select **Rename column**.



In the installed client, you can also right-click the column, and select **Sparkline settings**, **Calculated value settings**, **Icon settings**, or **Bullet graph settings** depending on the type of the dynamic item. Under **General**, change the **Name**.

## Hiding the column header for a dynamic item

You can decide to either hide the name of one column containing dynamic items or to hide the entire header row.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.



## Prerequisites

Make sure the graphical table you are working with is selected on the analysis page.

## Procedure

1. In the installed client, right-click the column header you want to hide and select **Sparkline settings**, **Calculated value settings**, **Icon settings**, or **Bullet graph settings** depending on the type of the dynamic items in this column.
2. Under **General** unselect **Show name in header**.
3. Click **Close**.

## Hiding the header row in a graphical table

Select the graphical table in the visualization properties panel and then select **Column headers: Hide all** under **Appearance**.

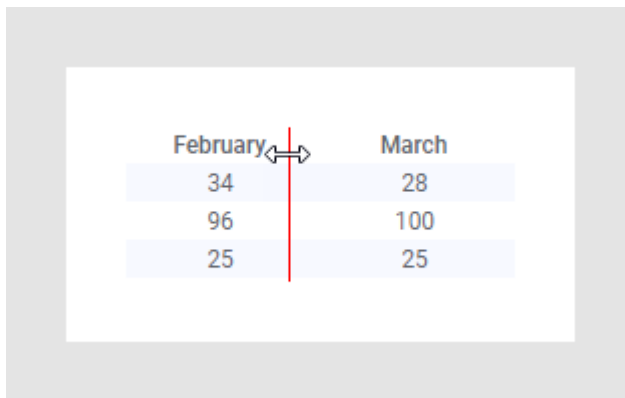
In the installed client, you can also right-click the graphical table and select **Properties**, then go to **Appearance** and unselect **Header row**.

## Changing height and width in tables

You can change row height and column width in tables and cross tables to better suit your content.

## Procedure

1. Place the cursor between two rows or two columns in the table, where the cursor changes appearance.
2. Adjust the row height or column width by clicking and dragging the red line that appears.



February	March
34	28
96	100
25	25



Press **Ctrl** simultaneously to adjust all the columns to the same width. The height of header row is adjusted separately, while the data rows always get the same height.

3. Release the cursor at the desired height or width.



To optimize the use of screen estate with all content in the columns kept visible, you can double-click the column divider to adapt the column width to the content. Alternatively, click the column header, and in the opened pop-up menu, select **Size to fit**.

You can also define an exact column width in pixels, by clicking the header and specifying a **Column width** in the pop-up menu.



In the installed client, you can optimize the use of screen estate for the entire table. Right-click the visualization, and select **Resize Columns to Fit**. This will make the column widths as small as possible. That is, every column adjusts automatically to the smallest size possible while keeping all content visible.

## Sorting the rows in a graphical table

You can sort the rows in a graphical table.

Click the header of the column you wish to sort the data by. You can sort by columns containing calculated values, icons (based on the underlying calculation), and bullet graphs (based on the underlying calculation of the values). Click the column header again to sort in descending order. Click once again to return to the original sort order.

If a hierarchy is used on the graphical table row axis, then sorting will be performed within each category only.

## Copying marked rows in a graphical table

You can mark rows in a graphical table and copy them.

Mark the rows of interest in the graphical table, then right-click them and select **Copy** from the pop-up menu.

Information about the marked rows is copied to the clipboard. For sparklines, the starting point and end point Y-axis values are copied. For calculated values, the actual values are copied. For icons, any applied rules are displayed on the form of the shape and the specified HTML color code. For bullet graphs, the value representing the bar is copied, and if a comparative value exists this is also copied.

## Making a column not respond to data limiting


The default behavior when adding a new column to a graphical table is that it uses the same data limiting, for example filtering, as the rest of the graphical table. You can, however, add a specific data limiting that only applies to one column.

See [Adding data limitations for a visualization](#) on page 555 and [Limiting the data in a visualization using filterings](#) on page 561 for more general information about limiting data in a visualization.



What is described here also applies to sparklines in the Text area.

### Procedure

1. In the visualization properties panel, below the visualization name, select the column that you want to exempt from data limiting.
2. At the bottom of the panel, click  **Add** and select **Item data limiting**.

3. In the expanded Item data limiting section, click  next to each data limiting to remove all of them.



You can also select one specific filtering scheme and not change any of the filters it contains.



In the installed client, you can also right-click the column that you want to exempt from data limiting and select **Sparkline settings**, **Calculated value settings**, **Icon settings**, or **Bullet graph settings** depending on the type of the dynamic items in this column. Select **Data** and under **Data limiting**, select **Specify separate data limiting**. Clear all check-boxes under **Limit data using markings** and **Limit data using filtering**, and clear the input field under **Limit data using expression** if applicable. Click **Close**.

Now the column always shows all data, regardless of any filtering or other data limitations in the other parts of the visualization.

## Adding or editing a rule for the color, shape, or font style of a dynamic item (icons, calculated values)

Icons need at least one defined rule for anything to be visible. It is the rules that determine what to display and when. Calculated values do not require any rules at all, but they can be used to emphasize a certain value using a color or a different font style.



Rules for dynamic items can only be authored in the installed client.



### Procedure

1. Right-click the column containing the dynamic items for which you want to add or edit a rule.

Option	Description
Icon	Select <b>Icon settings</b> and go to the <b>Icons</b> section.
Calculated values	Select <b>Calculated values settings</b> and go to the <b>Values</b> section.

2. Click **Add rule....** To edit an existing rule, right-click it and select **Edit rule....**

3. In the Add/Edit rule dialog, select the **Rule type** you want to apply.

Option	Description
Top	<p>Sets the top number of items to the selected appearance. Specify the number of items to affect in the Value field.</p> <div>  <p>In some cases more than the defined number of items may be affected by a rule. For example, if the rule states that the top three items should be yellow, and four items represent the highest value, then all four of these items will become yellow.</p> </div>
Bottom	<p>Sets the bottom number of items to the selected appearance. Specify the number of items to affect in the Value field.</p> <div>  <p>In some cases more than the defined number of items may be affected by a rule. For example, if the rule states that the bottom three items should be yellow, and four items represent the lowest value, then all four of these items will become yellow.</p> </div>
Between	Sets all the items representing values between the two specified values to the selected appearance. Specify the values in the two Value fields.
Equal to	Sets all the items representing values that are equal to the specified value to the selected appearance.

Option	Description
Not equal to	Sets all the items representing values that are not equal to the specified value to the selected appearance.
Greater than	Sets all the items representing values greater than the specified value to the selected appearance.
Greater than or equal to	Sets all the items representing values greater than or equal to the specified value to the selected appearance.
Less than	Sets all the items representing values less than the specified value to the selected appearance.
Less than or equal to	Sets all the items representing values less than or equal to the specified value to the selected appearance.
Boolean expression	Lets you define a boolean expression. All the items for which the expression is true will get the specified appearance.

4. Set the **Value** for the rule.
5. Specify the **Color** to use when the rule is applied. For the **Icon**, you can also select a **Shape**, and for the Calculated value a **Font style**, such as Regular, Bold, or Italic.
6. Click **OK**.



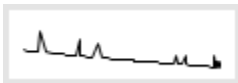
You can add more than one rule. If you add many rules, the rules are prioritized from top to bottom, for each row. None of the rules further down in the list will be considered once a rule higher up in the list has been applied. You can click and drag a rule in the Rules list to change the order.

## Sparklines

Sparklines are small, simple line graphs traditionally used for showing trends or variations of some variable.



Sparklines must be authored in the installed client.



They can be displayed in the [context of a graphical table](#) or in a text area. The general idea of sparklines is that they can be included directly where they are needed to provide context to a value.

Sparklines can be set up to change with filtering like any traditional visualization, or they can be locked to show fixed values using the settings in the Data properties for the Sparkline.

### Changing the scale for a sparkline column in a graphical table

You can choose if you want to use the same scale for all sparklines in the graphical table or a different one for each sparkline.

#### Prerequisites

Make sure the graphical table you are working with is selected on the analysis page.

#### Procedure

1. In the visualization properties panel, below the visualization name, select the sparkline for which you want to change the scale.

2. Locate the Y-axis section and expand it.
3. From the drop-down menu, select to use one **Single scale** for all sparklines in this column or **One scale for each sparkline**.

Use Single scale if you want to show the Y-axis values for all sparklines in this column presented using the same scale. Use One scale for each sparkline to maximize the Y-axis variation within each sparkline.



In the installed client, you can also right-click the sparkline column, and select **Sparkline settings**. On the **Axes** page, click to select whether to use **One scale for all sparklines in this column** or **Multiple scales**.

## Showing Y-axis starting point and end point values for the sparkline

You can display the Y-axis value of the start point and/or the end point of the sparkline.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

### Procedure

1. Right-click the graphical table and select **Sparkline settings**.
2. Under **Appearance**, select the **Starting point** and/or **End point** check box.
3. If you want, you can change how much space to use for the values by modifying the **Y-axes values width**.



This setting does not change the total width of the column. To increase the total column width, place the mouse pointer at the right edge of the column header you want to resize and adjust the width by holding down the mouse button and moving the mouse horizontally.

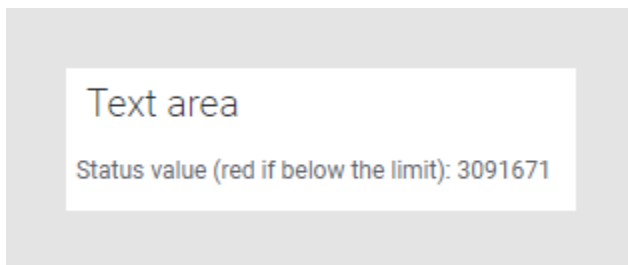
## Calculated values

Calculated values are values derived from some kind of aggregated expression, similar to the data shown in cross tables.

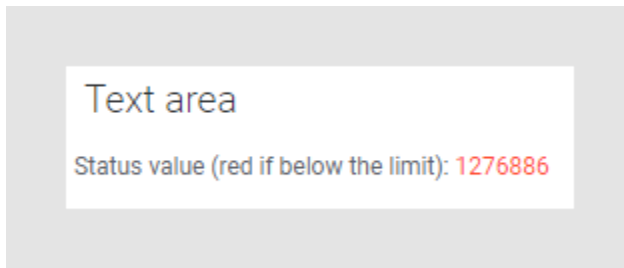


Calculated values must be authored in the installed client.

The general idea of calculated values is that they can be included directly where they are needed, in the [context of a graphical table](#) or a text area, to provide information at a glance.



By adding rules that control the color and font style you can make sure that a value stands out, when it falls outside of the specified limits:



Calculated values can be set up to change with filtering like any traditional visualization, or they can be locked to show fixed values using the settings in the Data properties for the Calculated value.

## Icons

Icons are small, simple images traditionally used for displaying trends or variations of some variable.



Icons must be authored in the installed client.

The general idea of icons is that they can be included directly where they are needed, in the [context of a graphical table](#) or a text area, to provide information at a glance. In the example below, icons are used in a graphical table to show the bottom, top, and intermediate sales region of some fictive product:

Region	Icon
Northeast	—
West	+
South	●

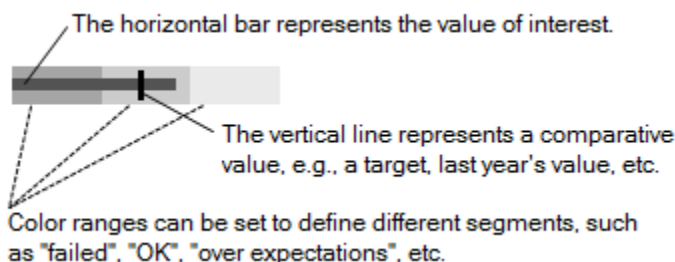
Icons can be set up to change with filtering like any traditional visualization, or they can be locked to show fixed values using the settings in the Data properties for the Icon.

## Bullet graphs

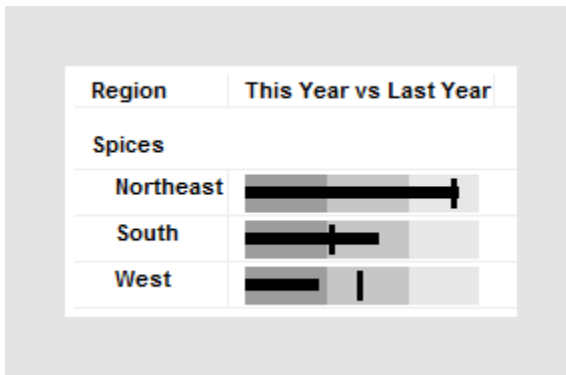
Bullet graphs are used to compare one value, represented by a horizontal bar, to another value, represented by a vertical line, and relate those to qualitative ranges.



Bullet graphs must be authored in the installed client.

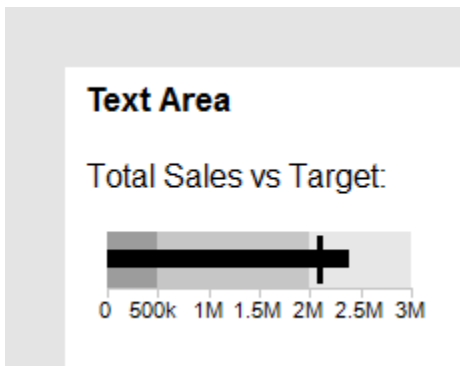


The general idea of bullet graphs is that they can be included directly where they are needed, in the [context of a graphical table](#) or a text area, to provide information at a glance. In the example below, the bullet graph resides in a graphical table:



It compares sales performance for spices in different regions this year to sales performance last year. The background color ranges are qualitative ranges. In the example above they may for example indicate that if the bar is within the leftmost color range, sales performance is poor, while if it is within the rightmost color range, sales performance is good.

In the text area, the bullet graph can be displayed with a scale, as seen below:



Bullet graphs can be set up to change with filtering like any traditional visualization, or they can be locked to show fixed values using the settings in the Data properties for the Bullet graph.


### Adding color ranges to a bullet graph

You can add color ranges to a bullet graph in order to define, for instance, within which range performance is considered to be good enough. In the example below, color ranges have been added to indicate that if the bar is within the leftmost range, then performance is below what is acceptable, but as long as the bar is within the rightmost range it is OK.



Bullet graphs must be authored in the installed client.

#### Procedure

1. Right-click the bullet graph column, and select **Bullet graph settings**.
2. In the **Color ranges** section, select the **Show color ranges** check box to enable the **Add** button, a color selector , and an empty text field. One single color range is displayed in the bullet graph.
3. Click the **Add** button to add another color range. A column selector is shown, and below it, another color selector with an empty text field.

4. In the column selector, define the value that should be the limit between the added color range and the one above it. You can choose a column directly from the column selector or you can right-click it to specify a [custom expression](#).



The start value of the first color range will automatically be the lowest value of the bullet graph scale, and the end value of the last color range will automatically be the highest value of the scale.



If you want you can use a specific value or percentage as the limit between the color ranges.


5. Repeat steps 3 and 4 for each color range you want to add.



A new color range is always added last in the list of existing color ranges in the dialog.

6. If you want, you can enter descriptive texts for the color ranges in the text fields. The texts can be shown in the tooltip for the bullet graph.

### Removing a color range from a bullet graph

1. Right-click the bullet graph column, and select **Bullet graph settings**.
2. In the **Color ranges** section, click  to the right of the color range you want to remove.

The color range is removed from the bullet graph, and the corresponding controls are removed from the dialog.

## Text area

A text area is a space where you can add information such as the purpose of a page, or maybe state the observations you have made, so that other people can verify or comment on your findings. You can also add images, links, and buttons, which can initiate actions that you think might be helpful for others who open your analysis.



Text areas must be authored in the installed client.

The text area is not a visualization as such, but it can be placed within a page just like a bar chart or scatter plot. For more information on how to position the text area in a page, see [Arranging visualizations](#) on page 603.

See also [Locking the size of certain visualizations](#) on page 623 to learn how to avoid scroll bars in text areas, when creating analyses that will be viewed on screens of different sizes.

There are several different types of content you can add to a text area:

- **Text** - text can be formatted to your liking, by changing the font, color, alignment, and so on. You can also add links leading to an external web page.
- **Images** - images can be added to the text area in GIF, BMP, PNG or JPG format. Regular images are added using **Insert Image**, but you can also add images that behave like action controls when clicking on them, see below.
- **Action controls** - you can add links, buttons or images that perform a certain [action](#), or series of actions, to the text area. For example, an action link can switch to a different page or apply a bookmark. It can also refresh a data function calculation or run a script. This can be handy if you intend to share your analysis with other people. You can, for example, write instructions in a text area, and include links to any operations you want them to perform, such as: "...and when you are done filtering, [refresh the calculation](#)". In addition, if any [reports](#) of the analysis have been prepared, they can be exported to a PDF document using an action control.



- **Property controls** - you can add a number of different items that control the values of selected [properties](#). It could be drop-down lists, list boxes or sliders with predefined values. You can also add manual input fields where anything can be entered, or you could simply add a label showing the current value of a specified property. See [Adding a property control to a text area](#) on page 325 for more information.
- **Filters** - if you only want to show a few filters in your analysis, you can add those filters to a text area. This means the filters panel and the data in analysis flyout are not needed for filtering purposes and can be closed to save screen estate. Filters in the text area can also be configured to use a different filtering scheme than the one used on the rest of the page. This could be useful if you have selected to limit one or more visualizations on a page by some other filtering scheme than the one used on the page.
- **Dynamic items** - dynamic items are small "visualizations" that can live within a text area or in a graphical table. When included in a text area they represent an aggregated view of some data. For example, this could be a calculated value showing the total sum of sales. The dynamic items can be configured to respond to the page filtering, some other filtering or no filtering. They can also be limited by markings in other visualizations, similar to details visualizations. The available dynamic items are [sparklines](#), [calculated values](#), [icons](#) and [bullet graphs](#).

## Adding a property control to a text area



By adding a property control to a text area (using the installed client), and populating it with values from a column, you can let other people control what to show in a visualization, even if they do not have authoring rights.

See [Configuring an on-demand data table](#) on page 61 for an example where the property control can be used to update an on-demand data table. The steps below uses this example as the basis for what to select.



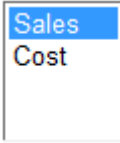
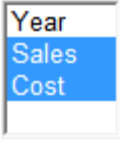

### Prerequisites

Text areas must be authored in the installed client.

### Procedure

1. Create or click to activate a text area.
2. On the title bar of the text area (or in the right-click menu) click **Edit Text Area** .
3. In the Edit Text Area dialog, type some descriptive text to help others understand what the control will do.
4. Click the **Insert Property Control** button , and select which type of control to add. In this example we will add a **Drop-down list**. See below for other options.

Option	Description
<b>Label</b>	Shows the current value of the selected property in the text area:  <b>Sales</b>  The label can also be used with binary properties. This means that you can use it to display images as well as text.
<b>Input field</b>	Adds a text box where you can type a new value for the property:  <b>Sales</b> <input type="text"/>
<b>Input field (multiple lines)</b>	Adds a text box where you can type new values for a string property with multiple lines:

Option	Description
	 <p>The <b>Input field (multiple lines)</b> control can only be used with string properties.</p>
Drop-down list	<p>Adds a drop-down list, where the property value can be changed by selecting among predefined options:</p> 
List box	<p>Adds a list box, where the property value can be changed by selecting among predefined options:</p> 
List box (multiple lines)	<p>Adds a list box, where the property value can be changed by selecting several predefined options:</p>  <p>The <b>List box (multiple select)</b> control can only be used with multiple line string properties.</p>
Slider	<p>Adds a slider where the property values can be changed by moving the slider:</p>  <p>The current property value is displayed in a label above the slider.</p>

- In the Property Control dialog, select a document property. In this example we use a string property called "Type", with the value "Apples".



You can also create a new property by clicking **New** and defining a name, type and a value.

- Select **Set property value through** and choose an option. In this example we use **Unique values in column**.

This option will show all unique values in the column as possible choices to select in the drop-down list. See below for other options.

Option	Description
Column selection	<p>Shows a list of columns to select from.</p> <p>If you not all columns should be available, you can type an expression to only show columns of interest. For example, use <code>datatype:integer</code> or <code>datatype:real</code> to only show columns of a certain data type.</p> <p>You can also search for columns matching a specific column property value. For example, if you have added a column property called <i>included.column</i>, with the default value <code>false</code>, and you update the value for the</p>

Option	Description
	columns that you want to include to true, you can use <code>included.column:true</code> in your search expression.
Unique values in column	Shows a list of unique values picked from a column.
Expressions	Shows a list of values obtained via more or less complex calculations.
Fixed values	Shows a list of values that you define yourself.
Numerical range	Shows a range of values from min to max, increased using a specified value interval.

- Depending on what you chose in step 6, you see different things here, but this example assumes that you chose Unique values in column. If more than one data table is available in the analysis, select the **Data table** to work on.

If you want to use the control to define an on-demand data table, then this should be the main data table and not the on-demand data table.

- Select **Column** to use. In this example, we use the "Type" column.
- If desired, you can limit the values to show in the drop-down list using a search expression. Only those values matching the search expression will be shown in the control. See Searching in Spotfire for more information about valid search expressions.
- Click **OK** and, in the Edit Text Area dialog, click **Save** before closing edit mode.

## Result

The property control is added to the text area. You can now use the property control to change the value of the property, and if the property is used on an axis, this can let you choose which product type to look at in a visualization.


## Adding a dynamic item to a text area

You can add a dynamic item to a text area, such as a sparkline, a calculated value, an icon, or a bullet graph. They are usually included in a graphical table and when they are used in a text area, they behave slightly differently. For example, you will not get instant feedback to your setting changes because the dynamic items sometimes do not show up until you leave the edit mode. The placement of the edit box tells you where in the text area the item will show up.




Text areas must be authored in the installed client.

## Procedure

- On the text area title bar, click **Edit text area** .



You can also right-click the text area and select **Edit text area** from the pop-up menu.

- Type a descriptive text in the text area to help others understand what the dynamic item will show.
- Click **Insert dynamic item** , and select **Sparkline**, **Calculated value**, **Icon**, or **Bullet graph**. The corresponding settings dialog opens.
- Make the necessary changes to the settings. See below for specific settings for each of the dynamic items.

Option	Description
Sparkline	In the Axes sections, select the column or hierarchy to define the <b>X-axis</b> and <b>Y-axis</b> of the sparkline.

Option	Description
	If you want, you can make other changes to the default sparkline settings, such as <a href="#">changing the filtering</a> in the <b>Data</b> section, <a href="#">adding an action</a> , or <a href="#">showing starting point and end point Y-axis values</a> .
<b>Calculated value</b>	<p>In the <b>Values</b> section, select a column, hierarchy, or expression in the <b>Calculated values using</b> field.</p> <p>If you want, you can make other changes to the default settings, such as <a href="#">changing the filtering</a> in the <b>Data</b> section, <a href="#">adding an action</a>, or <a href="#">setting up a rule</a>.</p>
<b>Icon</b>	<p>in the <b>Icon</b> section, select a column, hierarchy, or expression in the <b>Calculate icons using</b> field.</p> <p><a href="#">Define one or more rules</a> to control what icons to show and when.</p> <p>If you want, you can make other changes to the default settings, such as <a href="#">changing the filtering</a> in the <b>Data</b> section, or <a href="#">adding an action</a>.</p>
<b>Bullet graph</b>	<p>In the <b>Bullet graph</b> section, in the <b>Calculate values using</b> field, select a column or expression to define the horizontal bar in the bullet graph.</p> <p>Select a column or expression in the <b>Calculate comparative values using</b> field, to define the vertical line in the bullet graph. If you do not want to use a comparative value, select <b>Remove</b> in the column selector.</p> <p>If you want, you can change the <b>Color</b> of the bar and/or vertical line.</p> <p>You can make other changes to the default settings, such as <a href="#">changing the filtering</a> in the <b>Data</b> section, <a href="#">adding color ranges</a>, or <a href="#">adding an action</a>.</p>

- Click **OK** to add the dynamic item to the text area. Exit edit mode to see the result.

## Changing the size of a dynamic item in a text area

You can change the size of a dynamic item in the text area.




Text areas must be authored in the installed client.



To change the size of a calculated value, see [Changing the font, style and size of calculated values in a text area](#) on page 329.

### Procedure

- On the title bar of the text area where the dynamic item is located, click **Edit text area** .
- The outline of the dynamic item is shown as a gray box. Click the gray edit box to select it and drag either of the corners to resize the box.



[Only applicable to icon.]

The size slider within the icons settings can only resize the icon within the current edit box. If the slider is set to Max, then the edit box limit will be the limit of the icon. If the slider is set to Min, then the icon will be small, even if the edit box is made larger. Therefore, it is recommended that the slider in the Settings dialog is used only for fine tuning the size of really small icons.

- Exit edit mode to see the result.



You can always reset the size of the icon to the original size. Open the Format control dialog by right-clicking the gray box, click the Reset button, and then click OK.


## Changing the font, style and size of calculated values in a text area

You can change the font, style and size of calculated values in a text area.



Text areas must be authored in the installed client.

### Procedure

- On the title bar of the text area where the dynamic item is located, click **Edit text area** .
- Right-click the calculated value and select **Format control** from the pop-up menu.
- Adjust the font settings in the dialog to change the appearance of the calculated value to your liking.




The text color cannot be set here, it is [defined by rules](#).

- Click **OK** to apply the new font settings to the calculated value.
- Exit edit mode to see the result.



When working with calculated values, make sure you do not unintentionally override any font style settings that have been defined in a [rule](#). If you apply a font style (Bold and Italic) from the Format Control dialog, it will be added to the calculated value on top of any font style included in a rule for the calculated value. For example, if you select the check box for Bold in the Format Control dialog, then the calculated value will always be displayed as bold. A rule stating that the value should be displayed in bold only if the value is greater than a certain value will have no effect, since the calculated value is already displayed in bold.




## Edit text area toolbar


















The text area can only be edited when **Edit Text Area**  has been clicked on the title bar, or, when **Edit Text Area** has been selected from the pop-up menu.







Text areas must be authored in the installed client.

At the top of the opened Edit Text Area dialog, you will see a toolbar where a number of options are available:


Option	Description
	Cuts the selected object from the current position, to be pasted somewhere else.
	Copies the selected object.
	Pastes the selected object in the text area.
Font	Specifies the font of the text.
Size	Specifies the font size of the text.

Option	Description
	Sets the selected text to boldface.
	Sets the selected text to italics
	Underlines the selected text.
	Opens the Color dialog where you can specify the text color.
	Aligns the selected paragraph to the left of the text area.
	Aligns the selected paragraph to the center of the text area.
	Align the selected paragraph to the right of the text area.
	Creates a numbered list of the selected paragraph.
	Creates a bulleted list of the selected paragraph.
	Removes the indent characters from the selected text.
	Indents the text of the selected paragraph.
	<p>Opens a dialog where you can browse to locate an image to insert.</p> <p>Including images can be very costly to memory, so it is recommended to keep images small.</p>
	<p>Opens the Insert Web Link dialog where you can type or paste the link to any website which might be of interest for the analysis.</p> <p>You can also use this button to add a <code>mailto</code> link opening a new message in your standard email client, or use the Spotfire protocol handler to add a <a href="#">link to another analysis in the library</a>.</p> <p>Examples:</p> <pre>http://example.com</pre> <pre>mailto:person@example.com?subject=Test</pre> <pre>spotfire:server:https%3A//myspotfireserver/:analysis:84396295-4e71-2752-806b-ed80277af86e</pre>
	Removes the web link but leaves the text intact.
	Opens a drop-down menu where you can specify a background color for the text area.
	<p>Lets you specify a background image for the text area. Click <b>Browse for Background Image</b> to select the image and click <b>Background Image Settings</b> to change the positioning settings.</p> <div>  <p>Use your favorite image editor to make the image look the way you want it before adding it to the text area.</p> </div>

Option	Description
	Opens a dialog so you can select a filter to add to the text area.
	Lets you insert an action link, button, or image, which applies a bookmark, switches page, exports a report to PDF, runs a script, etc. See also <a href="#">Adding actions to a text area</a> on page 636.
	Lets you add a property control such as an input field, a drop-down list, a list box, and so on, for easily changing the value of a custom property. See also <a href="#">Adding a property control to a text area</a> on page 325.
	Allows you to insert a dynamic item (a <a href="#">sparkline</a> , a <a href="#">calculated value</a> , an <a href="#">icon</a> , or a <a href="#">bullet graph</a> ) into the text area.

To change the name of the text area, or to show or hide the title bar, go to the visualization properties.

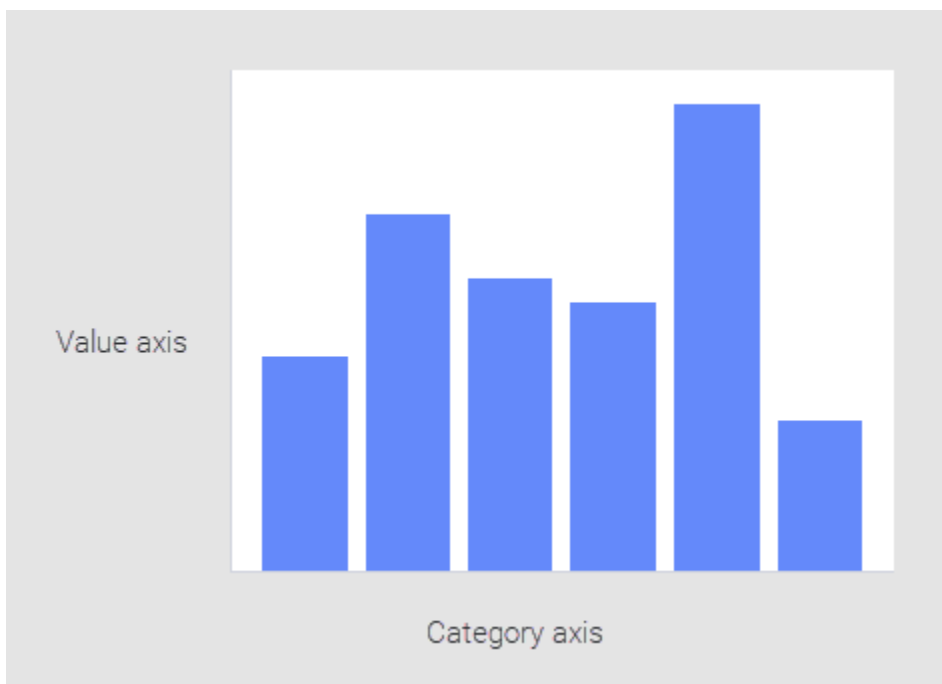


If you want to manually change the underlying HTML or add JavaScript code, click the Edit HTML button, . However, if you edit the HTML for a text area directly it is recommended to continue using this method for any further editing of the text area. This is because the standard WYSIWYG editor might add or remove changes to the HTML that you have not expected, depending on its `PerformHtmlSanitation` preference setting. The HTML sanitizer is switched on (`true`) by default. The [Edit HTML](#) dialog will provide you with some information regarding unsupported elements which you will only see in this mode. See also [Supported HTML in the Text Area](#).

## Bar chart

In a bar chart, you can compare values for different categories in your data (continuous data can be made categorical by auto-binning).

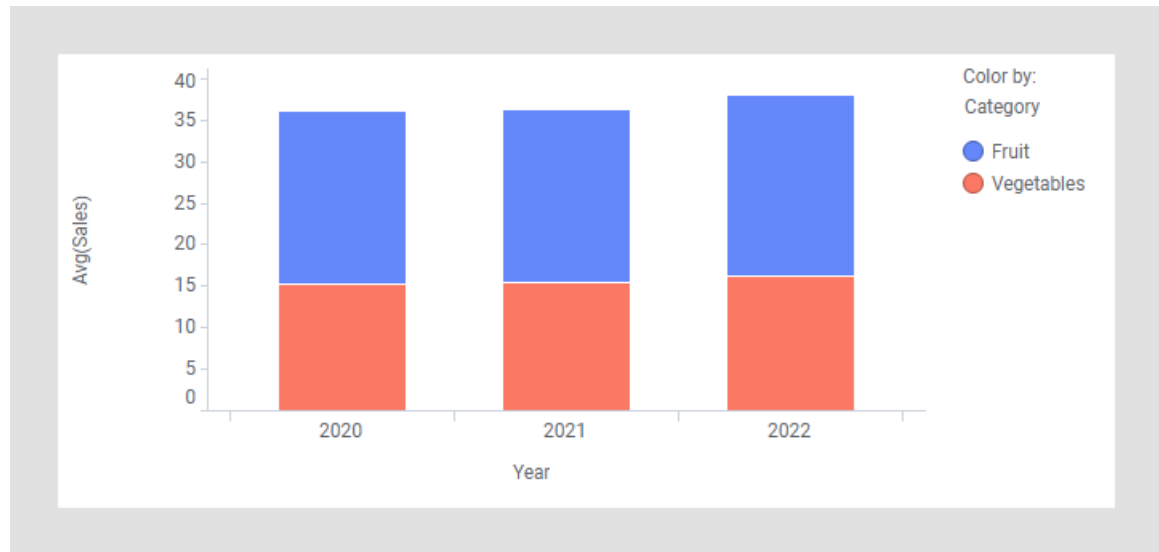
There are two types of axes in the bar chart, a **Category** axis and a **Value** axis. In most cases, columns containing numerical data are selected on the **Value** axis, and columns with categories are selected on the **Category** axis.



The bar chart shows data using a number of bars, each representing a particular category. The height of each bar is proportional to a specific aggregation (for example the sum of the values in the category it represents). The categories could be something like an age group or a geographical location. It is also possible to color or split each bar into another categorical column in the data, which lets you to see the contribution from different categories to each bar or group of bars in the bar chart.

### Example

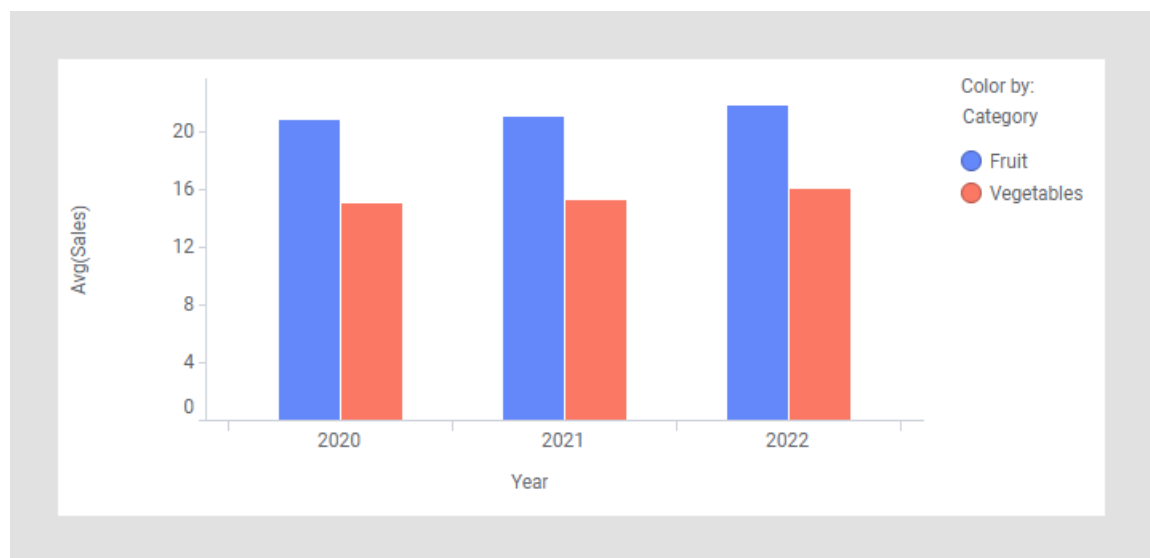
The bar chart below shows the total sum of sales for fruits and vegetables over three years.



The bars are colored by the Category column, which contains two values: Fruit and Vegetables. You can also view these color categories side-by-side as in the bar chart below. Locate the layout section in the visualization properties and click the **Side-by-side** option.



In the installed client, you can also right-click in the visualization and select **Side-by-Side Bars** from the pop-up menu directly.



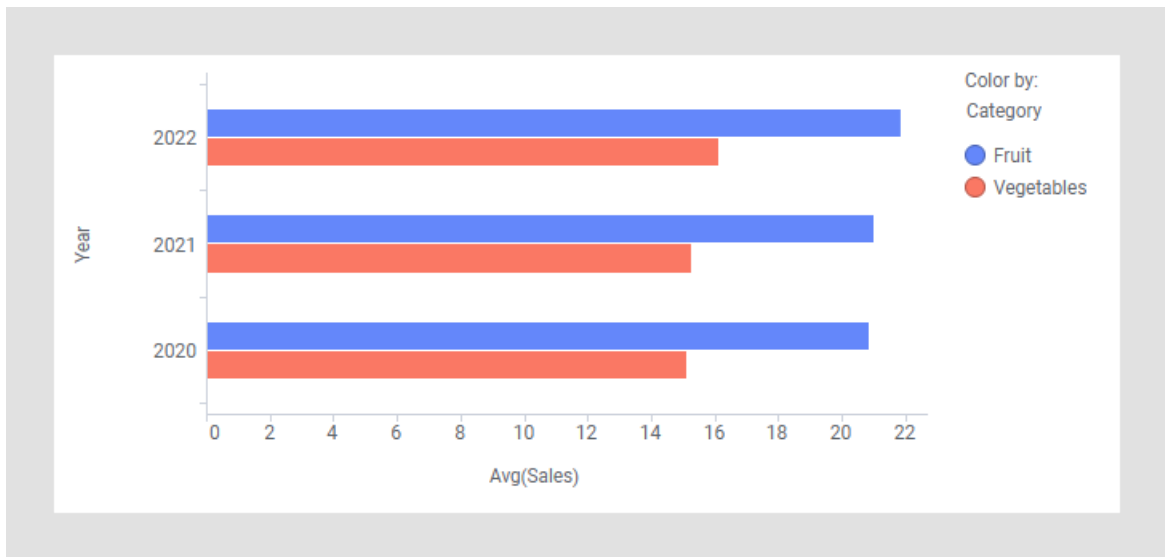
Another way to get a similar appearance is to add the column Category to the category axis to create a hierarchy on that axis. See [Hierarchies](#) to learn more about multiple columns on an axis.

If you prefer the bars to be displayed horizontally, locate the bar orientation section in the visualization properties and click the **Horizontal** option.





In the installed client, you can also right-click in the visualization and select **Horizontal Bars** from the pop-up menu directly.



Note that in some cases the category axis is referred to as X-axis, and the value axis is referred to as Y-axis. This is the case when adding some lines and curves, for example, as well as in [OVER statements in custom expressions](#).


All visualizations can be configured to show data limited by one or more markings in other visualizations only (details visualizations). Bar charts can also be limited by one or more filterings. Another alternative is to configure a bar chart without any filtering at all. See [Adding data limitations for a visualization](#) on page 555 for more information.

## Creating a bar chart

In a bar chart, you can compare values for different categories in your data.

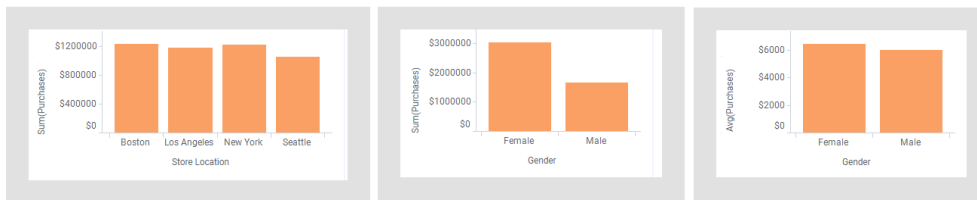
The heights of the bars represent a certain aggregated value for each of the categories. An aggregated value could be, for example, a sum or an average.

### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Bar chart** visualization type to the wanted position on the analysis page.  
A suggestion of a bar chart is presented.
3. On the **Value** axis, [select the column](#) whose values you are interested in comparing.
4. [Select which type of aggregated value](#) you want to show.  
The heights of the bars are recalculated to reflect your selected aggregation.
5. On the **Category** axis, select the column containing the categories by which you wish to split the data.

## Examples

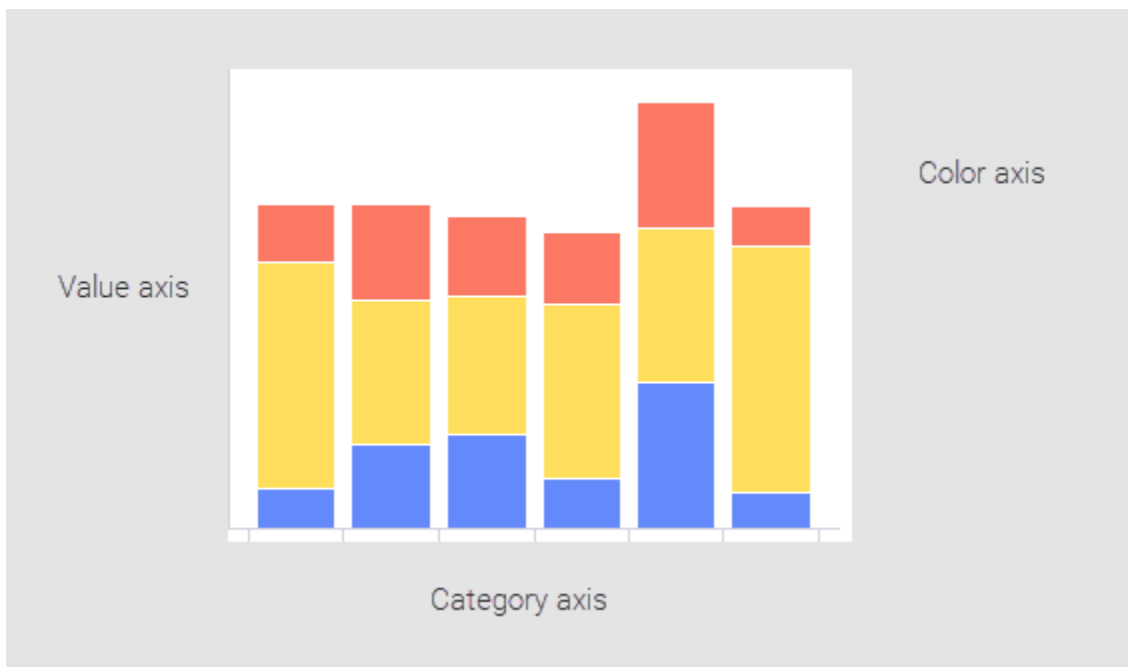
The lefthand and middle bar chart show the total sum of purchases but split into different categories on the Category axis. In the rightmost bar chart another aggregation value, average, is set.



See also [Creating a stacked bar chart](#), [Creating a side-by-side bar chart](#), and [Creating a 100% stacked bar chart](#).

## Creating a stacked bar chart

In a stacked bar chart, the bars are split into colored bar segments placed on top of each other. The total height of a bar shows the numerical value for a certain category, and the heights of the bar segments represent how different components contribute to that value.

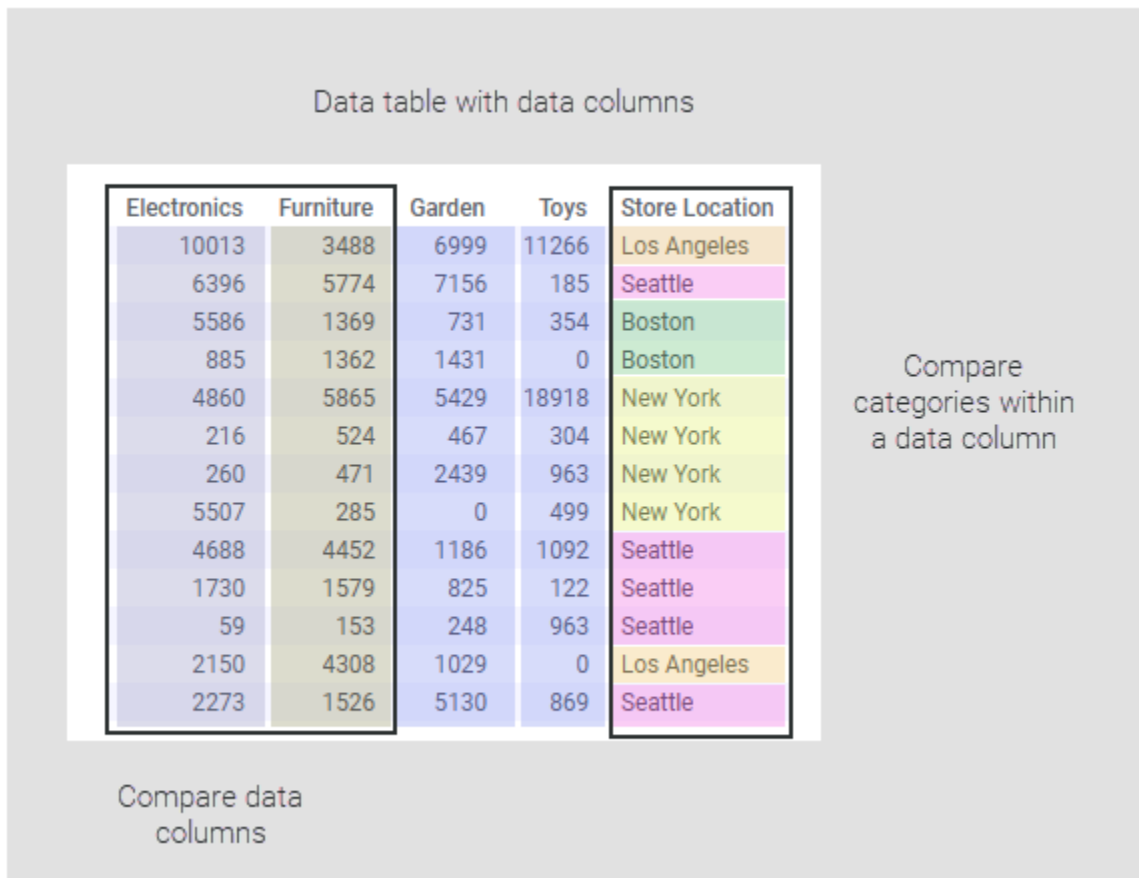


For information about the simplest bar chart, that is, a bar chart without colored segments, see [Creating a bar chart](#).

In a stacked bar chart, the **Color** axis in the legend is one of the main axes in addition to the **Value** and **Category** axes described in [Creating a bar chart](#). The **Color** axis is used to split the data in the different categories on the **Category** axis into further categories.

Another display option of bar charts with bar segments is a side-by-side bar chart, see [Creating a side-by-side bar chart](#). In a side-by-side bar chart, you can easily compare the heights of the bar segments, because they are placed next to each other.


The categories, or segments, into which a bar is split can be of different types as illustrated below: entire columns that you want to compare, or categorical values within a column.



Depending on what you want the bar segments to represent, you create a stacked bar chart in different ways.

## Comparing data columns

### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Bar chart** visualization type to the wanted position on the analysis page. A suggestion of a bar chart is presented.
3. On the **Value** axis, [select the columns](#) whose numerical values you are interested in comparing. The bars are split into stacked segments, colored differently. The **Color** axis in the legend is set to (Column Names), which means each selected column is represented by segments in a specific color.
4. For each of these selected columns on the **Value** axis, [select which type of aggregated value](#) to display.
5. On the **Category** axis, select the column containing the categories each bar should represent.

### Example

Assume the data table below is loaded. It contains electronics and toys purchases made by customers at different store locations. Gender is also registered.


Customer ID	Store Location	Gender	Electronics	Toys
SSMM55001	Los Angeles	Female	10013	11266
SSMM55002	Seattle	Female	6396	185
SSMM55003	Boston	Female	5586	354
SSMM55004	Boston	Male	885	0
SSMM55005	New York	Female	4860	18918
SSMM55006	New York	Female	216	304
SSMM55007	Los Angeles	Female	436	0
SSMM55008	New York	Female	0	0
SSMM55009	Los Angeles	Male	153	0
SSMM55010	Boston	Female	600	1034

See the settings made to examine differences in purchases made at different store locations. The total sums of electronics and toys purchases can be viewed as well as the electronics and toys contributions to the total sum at each location.



### Comparing categories within a column

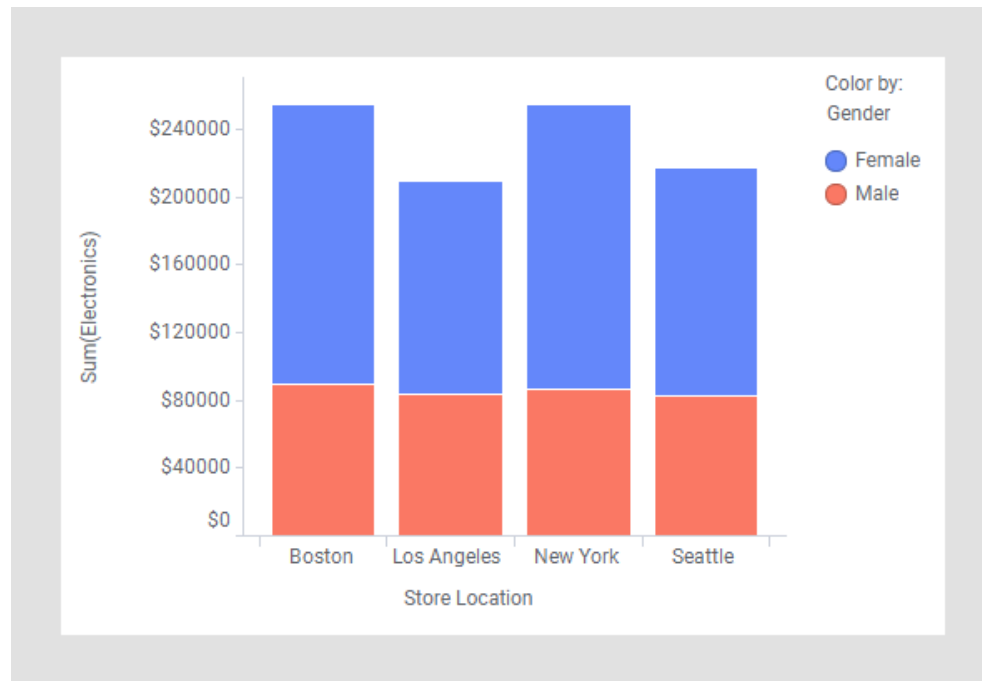
#### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Bar chart** visualization type to the wanted position on the analysis page. A suggestion of a bar chart is presented.
3. On the **Value** axis, [select the column](#) whose numerical values you are interested in comparing.

4. [Select which type of aggregated value](#) to display for the selected column.
5. On the **Category** axis, select the column containing the categories each bar should represent.
6. On the **Color** axis in the legend, specify the column containing the categories the bar segments should represent.

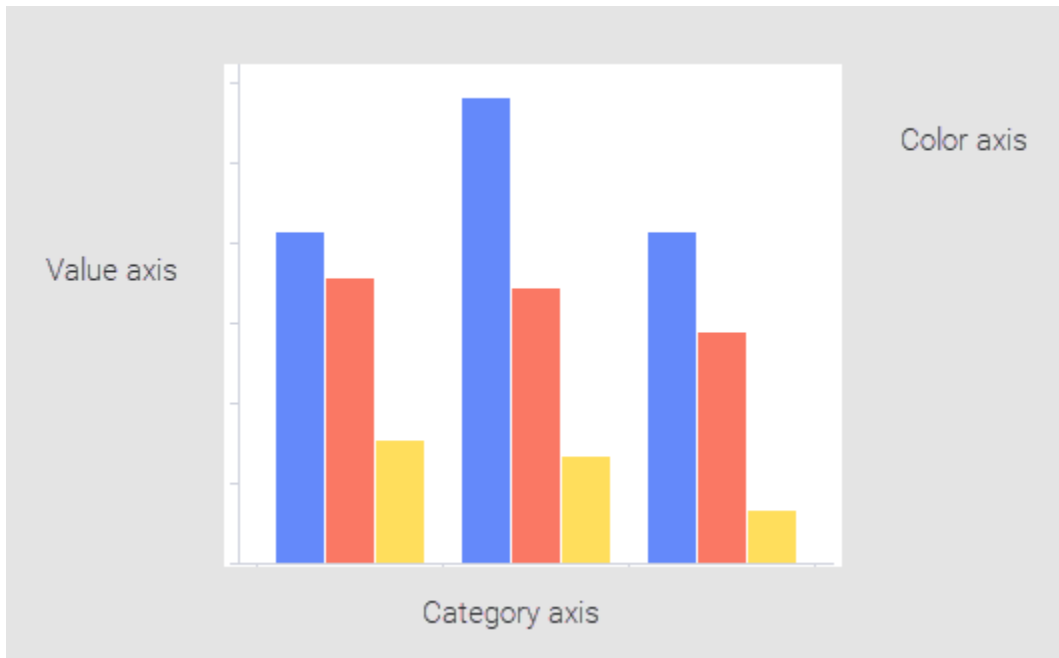
### Example

Assume the data shown in the example above is loaded. See the settings made to display electronic purchases made by men and women, respectively, at different store locations.



## Creating a side-by-side bar chart

In a side-by-side bar chart, the bars are split into colored bar segments. The bar segments are placed next to each other.



The bars in a bar chart can be split into colored segments as described in [Creating a stacked bar chart](#). In a stacked bar chart, the bar segments within a category bar are placed on top of each other, and in a side-by-side bar chart, they are placed next to each other. Because they are placed next to each other you can easily compare their heights.

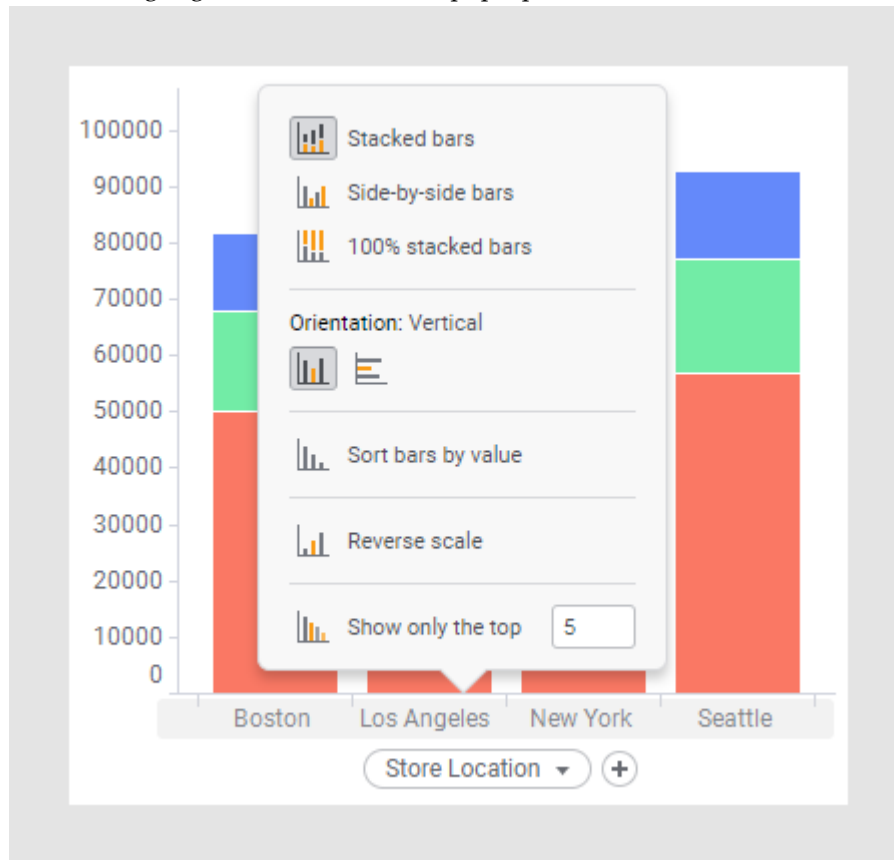
### Prerequisites

A stacked bar chart is created.

### Procedure

1. Place the cursor on the **Category** axis.  
The axis is highlighted.

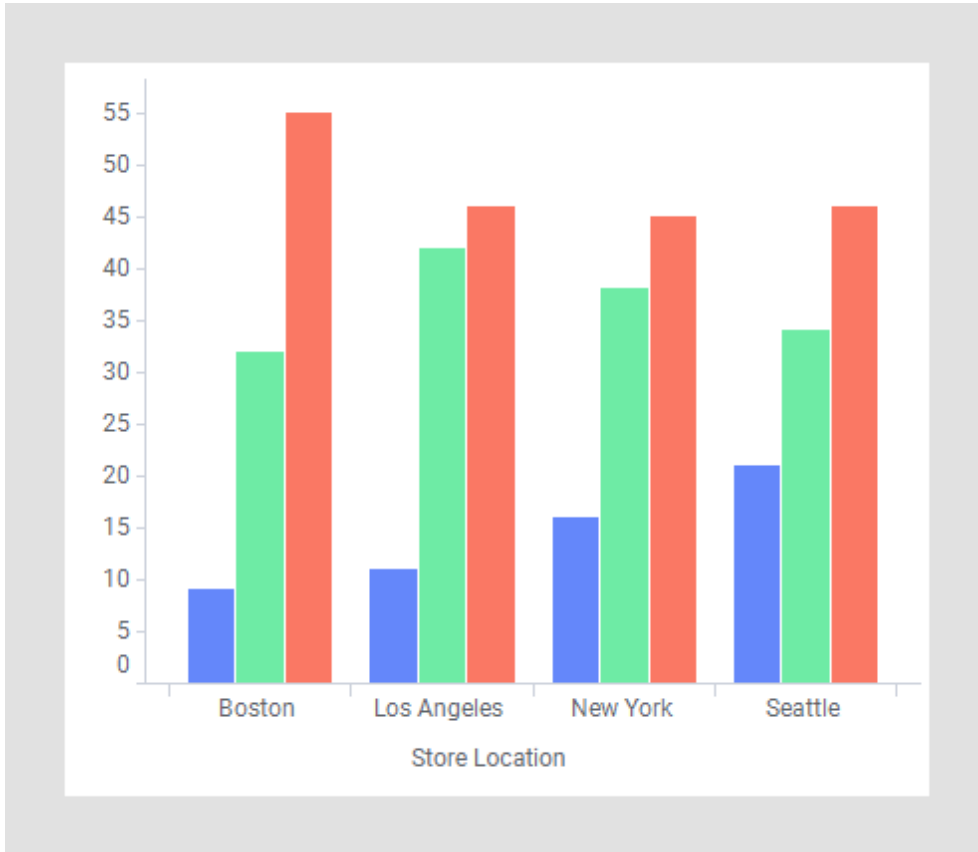
2. Click the highlighted area to access a pop-up menu.



3. Select **Side-by-side bars**.

### Example

The bar chart below shows the side-by-side bars display option of the bar chart in the image above. Note that the scale of the **Value** axis adapts to the heights of the bar segments.

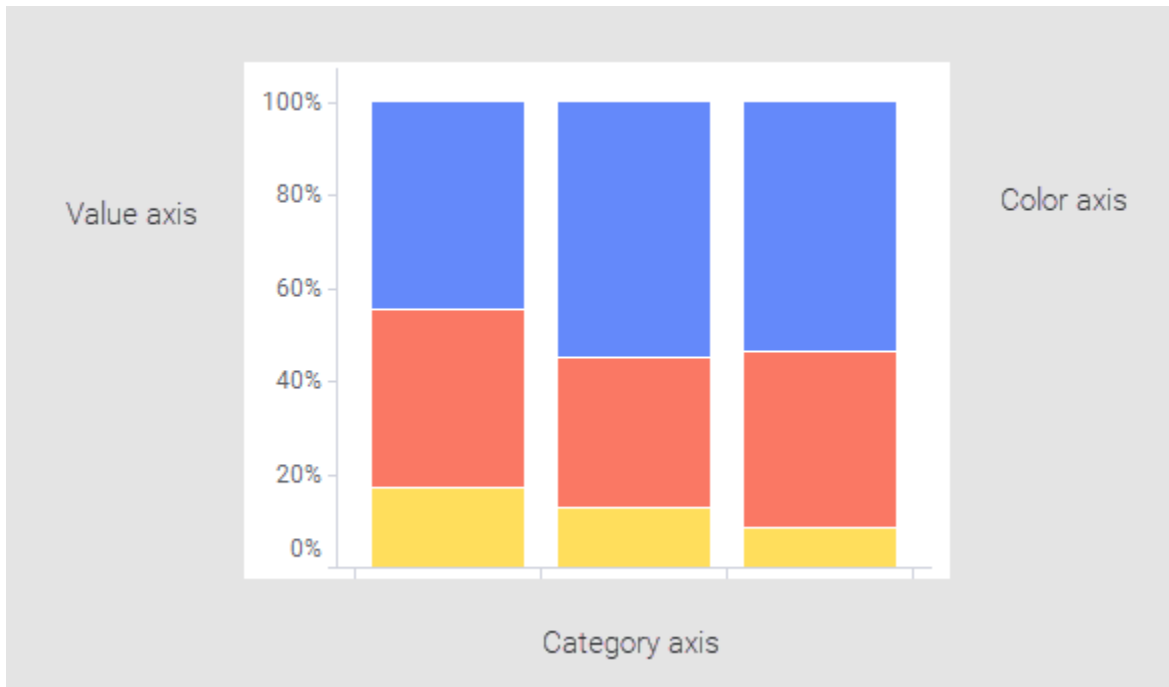


You can turn a side-by-side bar chart into a stacked bar by selecting **Stacked bars** in the pop-up menu.



## Creating a 100% stacked bar chart

In a 100% stacked bar chart, the bars are split into colored bar segments placed on top of each other. Each bar height is 100%, and the colored bar segments represent the components' relative contributions to the total bar.



In the common stacked bar chart, described in [Creating a stacked bar chart](#), the absolute values of the components make up the bars, which result in different bar heights. The 100% stacked bar chart differs from the common stacked bar chart by expressing the values of the components as percentages of the entire bar as shown above. This way the differences between the components are more put on view; not the actual values.

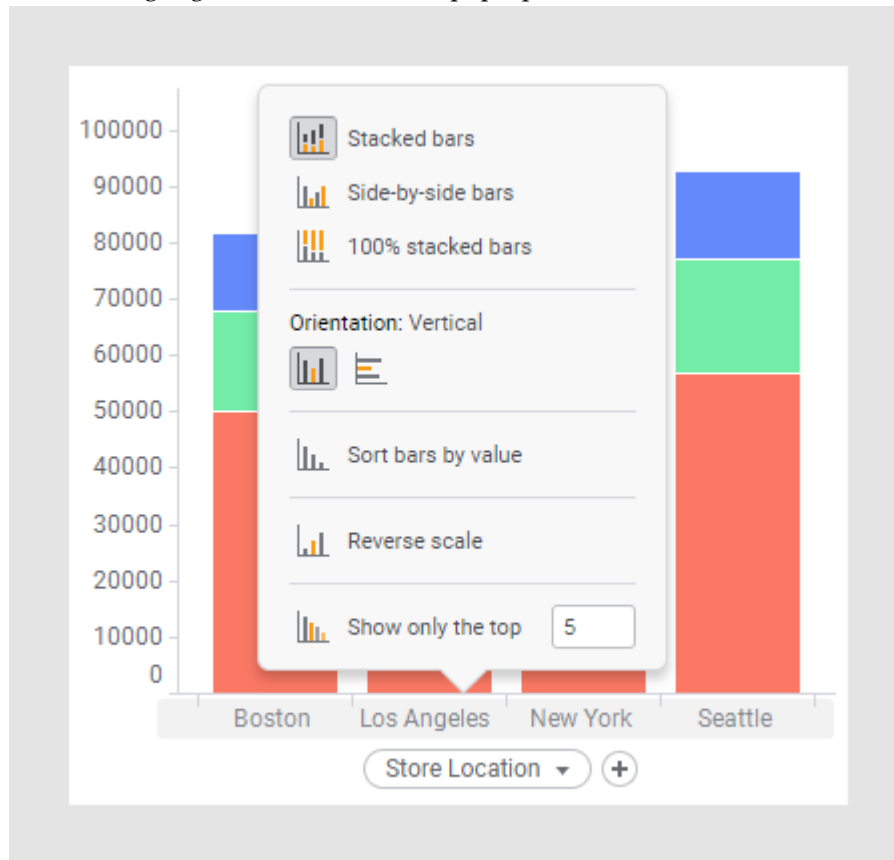
### Prerequisites

A stacked bar chart is created.

### Procedure

1. Place the cursor on the **Category** axis.  
The axis is highlighted.

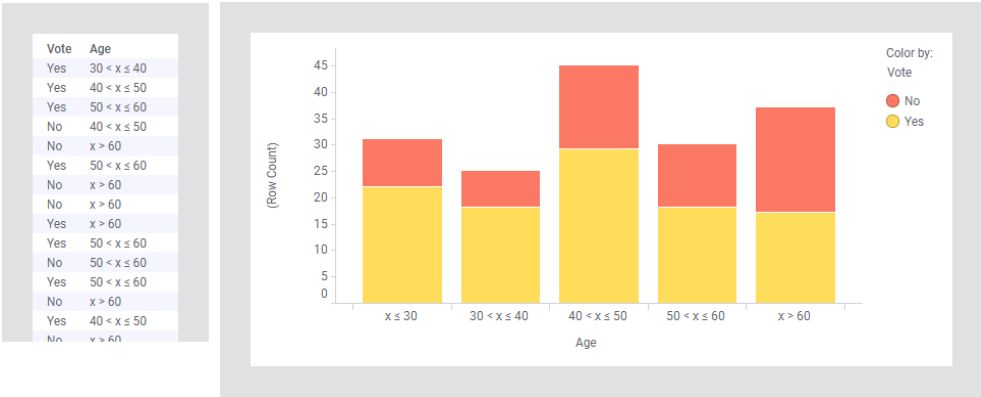
2. Click the highlighted area to access a pop-up menu.



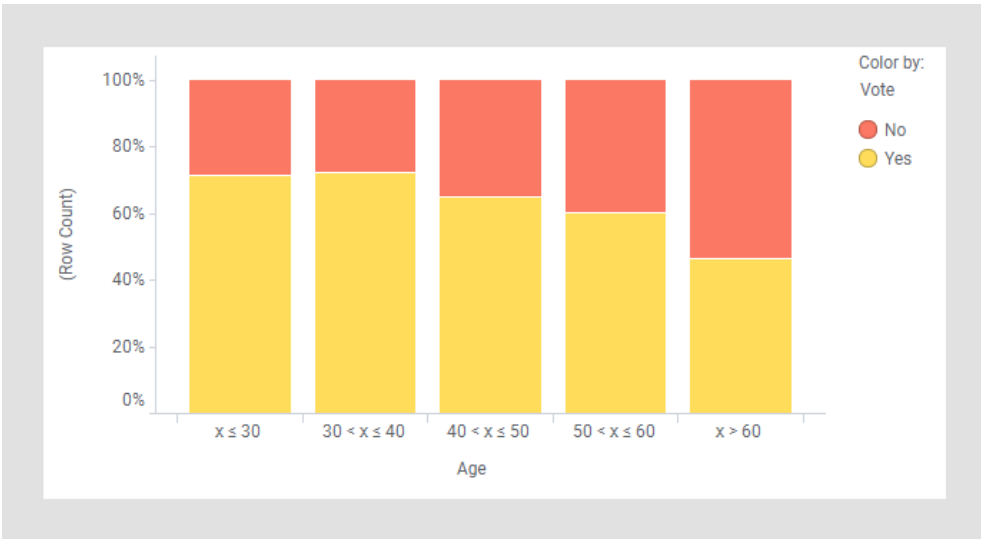
3. Select **100% stacked bars**.

### Example

Those taking part in a survey answered Yes or No to a question. Because there was an interest in examining any differences by cause of age, they were placed in age groups. How many answered Yes and No, respectively, is shown in the stacked bar chart split per age group.



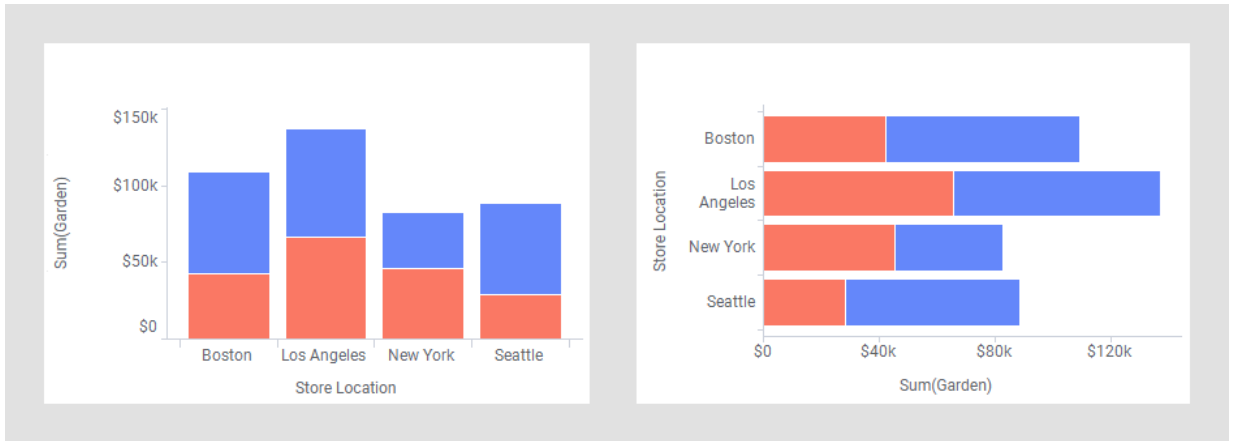
Changing the bar chart to a 100% stacked bar chart shows the Yes and No components' relative contributions to each bar.



The trend seems to be that the older participants, the more they answered No.

## Creating a horizontal bar chart

The bars in a bar chart are by default vertical. You can change the vertical orientation of the bars to horizontal.



In a vertical bar chart, long names of column values on the **Category** axis can be cut off. This can be avoided if you switch to a horizontal bar chart, where long names can be split into more than one row. You can also place the cursor on the scale line and, when the cursor appearance changes, drag it to make room for long names.

### Prerequisites

A vertical bar chart is created.


You can switch between a vertical and horizontal orientation anytime, that is, before, meanwhile, or after adjusting the bar chart properties.

### Procedure

1. Place the cursor on the **Category** axis.  
The axis is highlighted.

2. Click the highlighted area to access a pop-up menu.



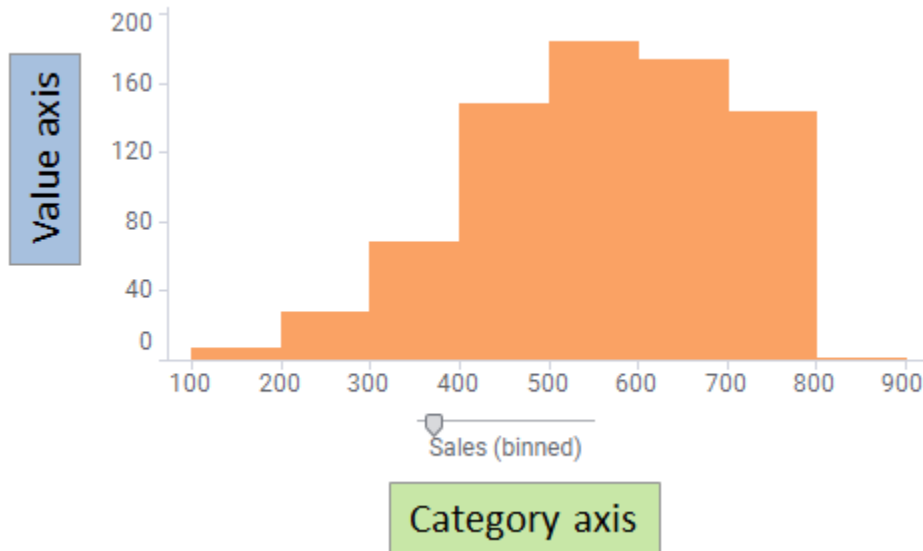
3. Beneath **Orientation**, click .



If you want to change the horizontal orientation of the bar chart to vertical, repeat step 1-2 above, and then, beneath **Orientation**, click .

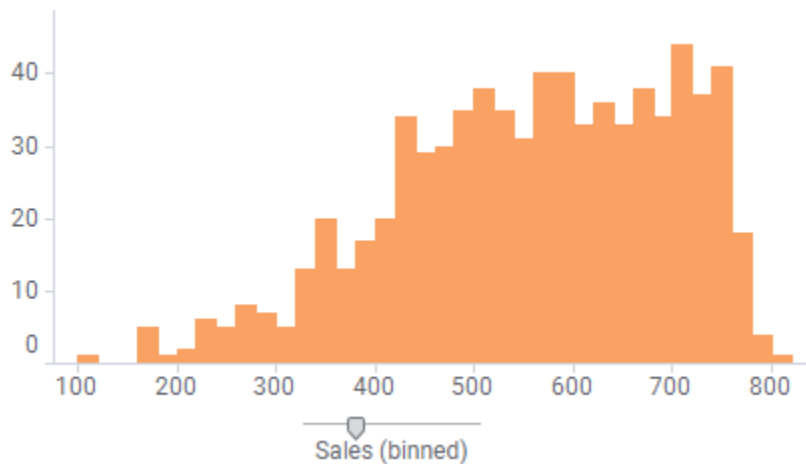
## Creating a histogram

In a histogram, you can show the distribution of numerical data. The entire range of the numerical values is divided into equal intervals on the Category axis, and for each interval, it is indicated on the Value axis how many individual data values that fall within it.




The histogram above is based on a bar chart. It shows how sales figures are distributed between the lowest and the highest amount. Usually there are gaps between each of the bars in a bar chart. In a histogram though, the bars are placed next to each other to better visualize the distribution of the numerical values along the range. Each interval of the range, also known as a bin, makes a category that is represented by a bar, and the height of a bar shows the number of data values in that particular interval.

The number of bins can be increased or decreased by dragging the slider on the axis. The following histogram shows the same data as the previous histograms, but the number of bins has been increased.



The steps below describe how to create a histogram based on a bar chart. A histogram based on a line chart is created in a similar way.

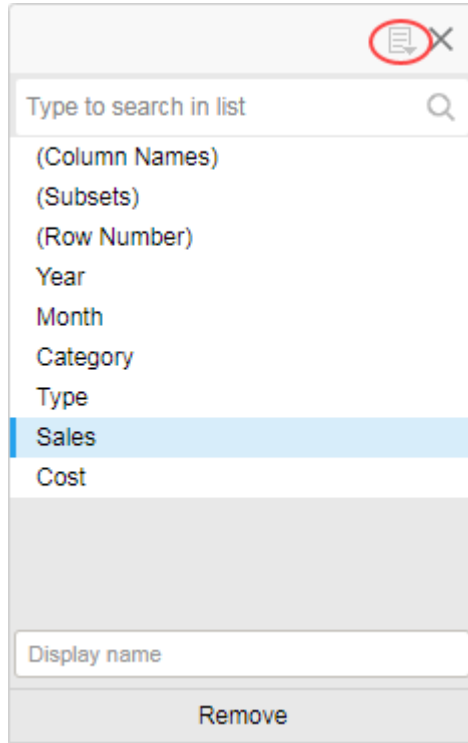
### Procedure


1. On the authoring bar, click **Visualization types**  to open the flyout.

2. Drag the **Bar chart** visualization type to the wanted position on the analysis page.  
A suggestion of a bar chart is presented.
3. On the **Category** axis, [select the numerical data column](#) for which the distribution of the values is of interest.

If the amount of data is big, the range of values on the axis is automatically divided into bins. If no automatic binning is made, do as follows:

- a) Click the column selector on the axis to open its popover.



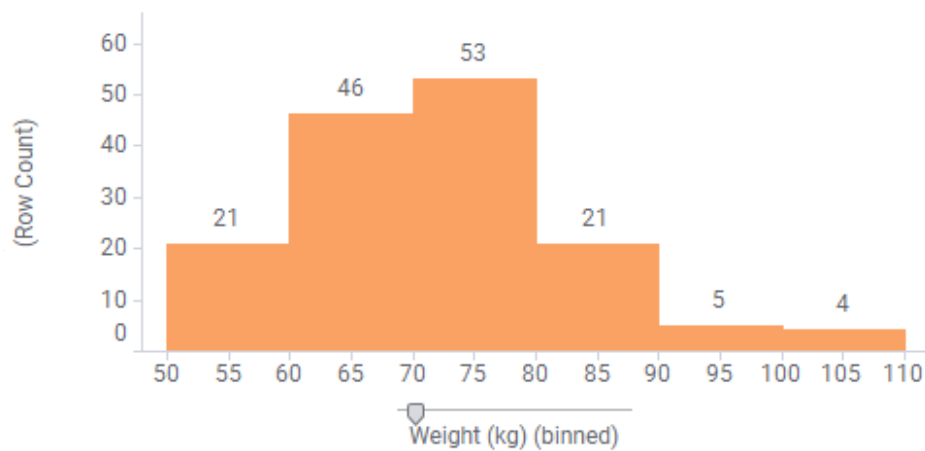
- b) In the upper-right corner of the popover, click  and then select **Auto-bin column**.  
The range of values on the axis is divided into bins. A slider where you can change the number of bins supplements the column selector.
4. Increase or decrease the number of bins by dragging the slider.  
The bars adjust to show how many values that fall within the specified bins.

### Example

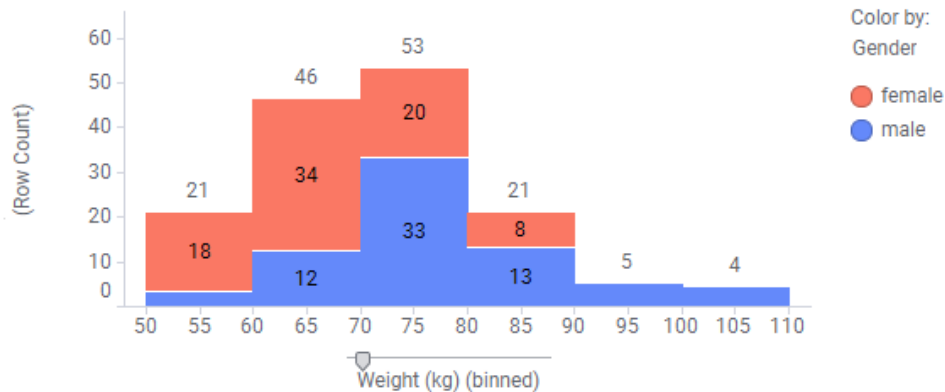
The table lists body weights for 150 men and women. The weights range from 50.0 to 110.0 kg.

Gender	Weight (kg)
female	62.70
female	54.00
female	64.20
male	63.20
male	70.40
male	72.80
male	76.80
female	61.60
female	70.40
female	66.80
female	58.80

The histogram below displays the distribution of values across 6 bins. The first bar shows the number of people that weighs between 50 and 60 kg, the second bar between 60 and 70 kg, and so on.



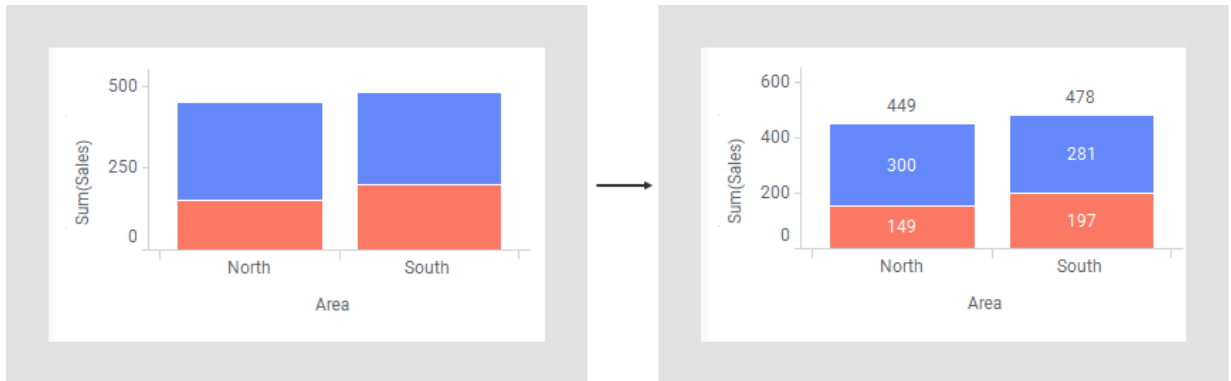
If you color by gender, you can examine male and female distribution differences as well.





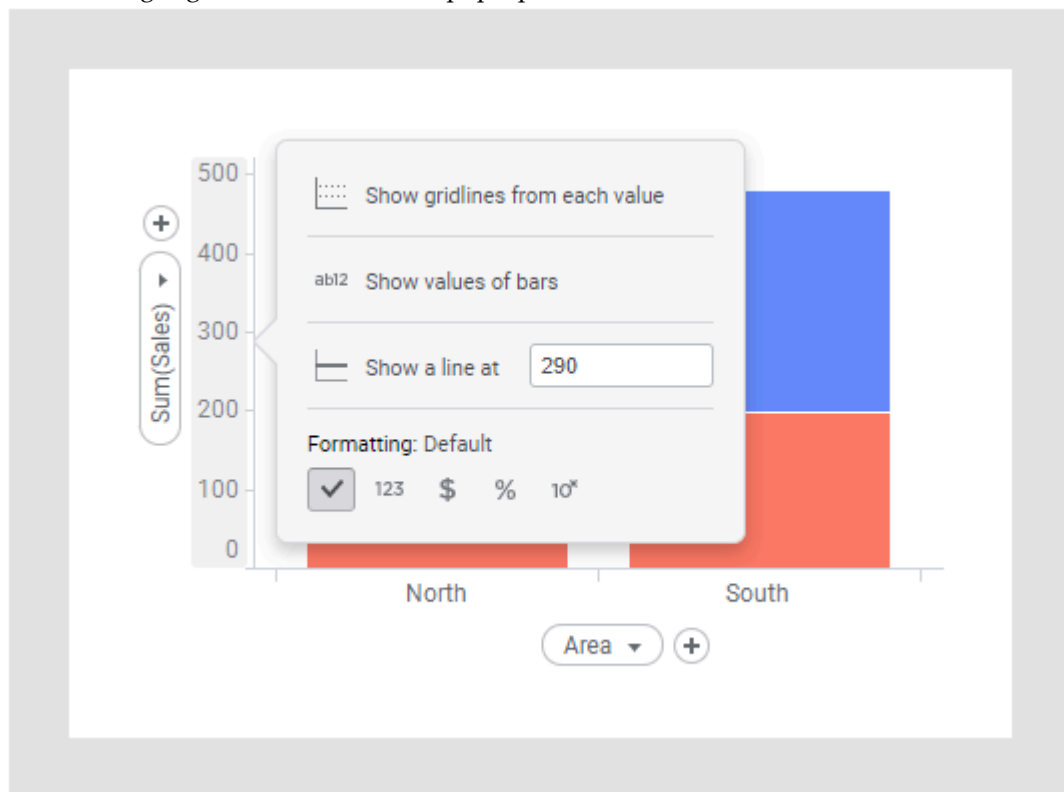
## Showing values of bars

You can show the values of entire bars and bar segments.



### Procedure

1. Place the cursor on the value axis.  
The axis is highlighted.
2. Click the highlighted area to access a pop-up menu.



3. Select **Show values of bars**.



If you want to remove the values from the bars, repeat the steps above.

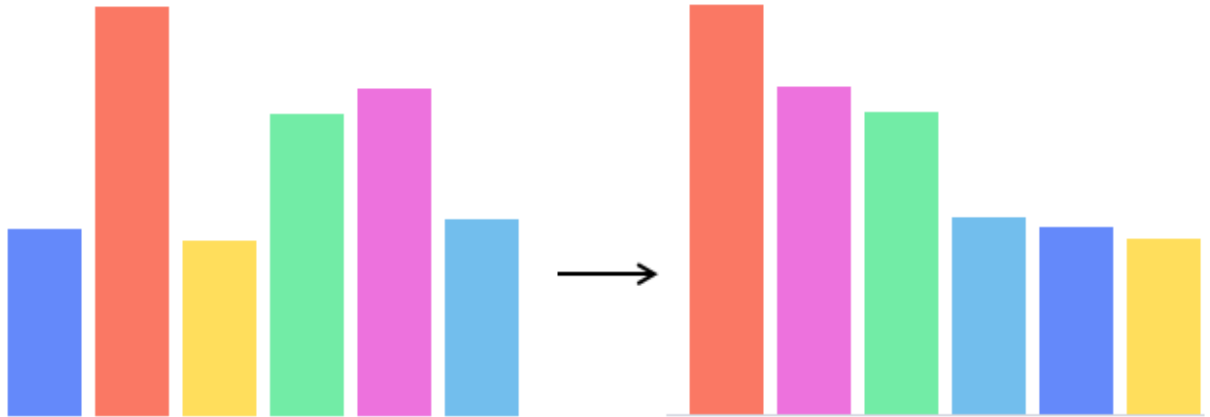


You can display values for complete bars only, for bar segments only, or display all these values. Right-click the visualization, select **Properties**, and click **Labels**. Beneath **Types of labels**, use the two check boxes, **Complete bar** and **Bar segments**, to specify which values to show.

## Sorting bars by value

You can sort the bars in a bar chart by height.

When you sort, the bars are sorted from the highest to the lowest bar.

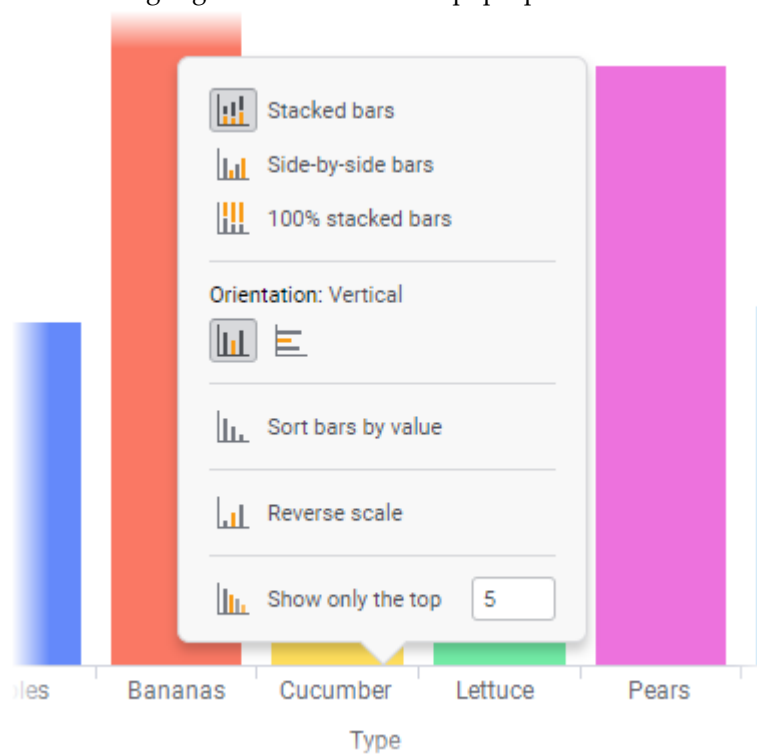


### Prerequisites

A bar chart is created.

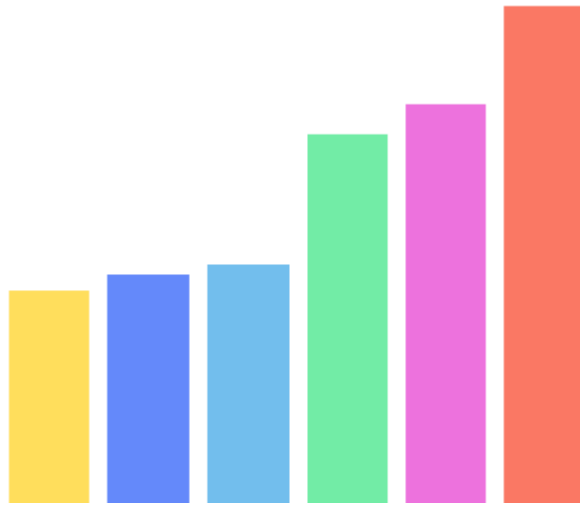
### Procedure

1. Place the cursor on the **Category** axis.  
The axis is highlighted.
2. Click the highlighted area to access a pop-up menu.



### 3. Select **Sort bars by value**.

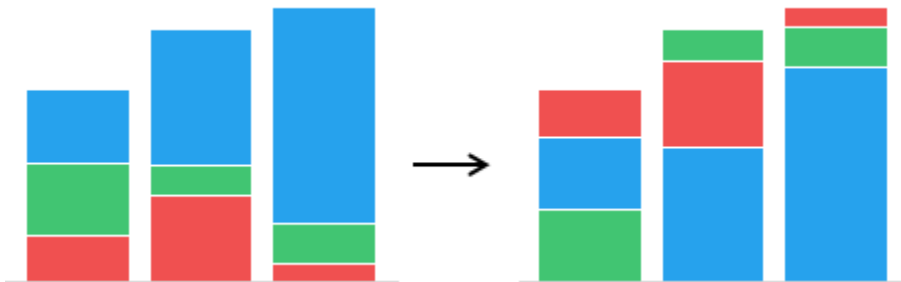
If you want to sort the bars from the lowest to the highest bar, select **Reverse scale** after clicking **Sort bars by value**.



## Sorting bar segments by value

You can sort the bar segments within stacked bars by height.

The highest segment within a bar is placed at the base, and the lowest segment at the top.



### Prerequisites

A stacked bar chart, or a 100% stacked bar chart, is created.

### Procedure

1. Right-click the bar chart, and select **Properties** in the opened menu.  
The Properties pop-over is shown.

- Click **Layout and sorting**.  
The **Layout and sorting** section is shown.

**Layout and sorting**

**Orientation**

☒ Vertical bars

☐ Horizontal bars

**Layout**

☒ Stacked bars

☐ Side-by-side bars

☐ 100% stacked bars

**Sorting**

☐ Sort bars by value

☐ Sort bar segments by value

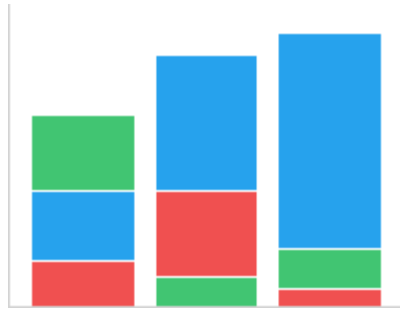
☐ Reverse bar segment order

- Select the **Sort bar segments by value** check box.

### Result

The bar segments are sorted by height within each bar.

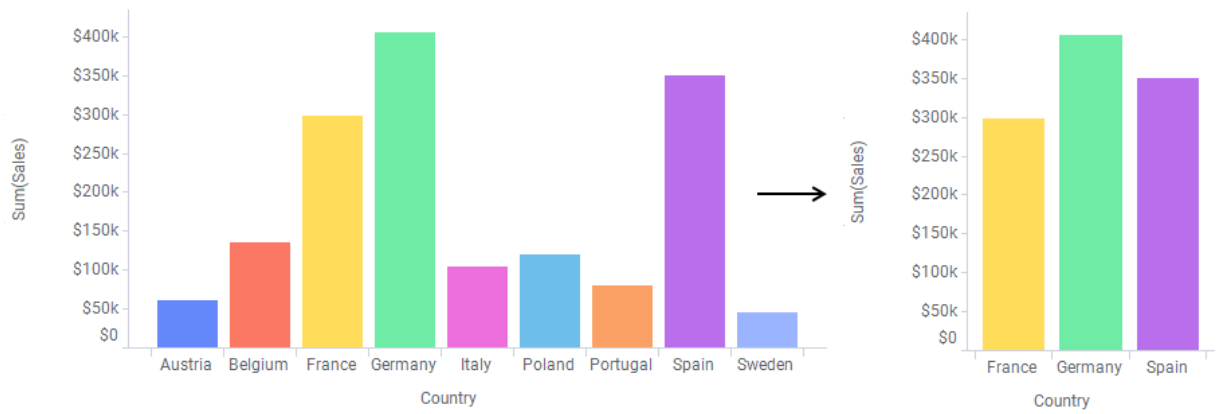
If you want to sort the bars the other way around, that is, place the lowest segment at the base of the bar and the highest segment at the top, select also the **Reverse bar segment order** check box.



### Showing only top bars

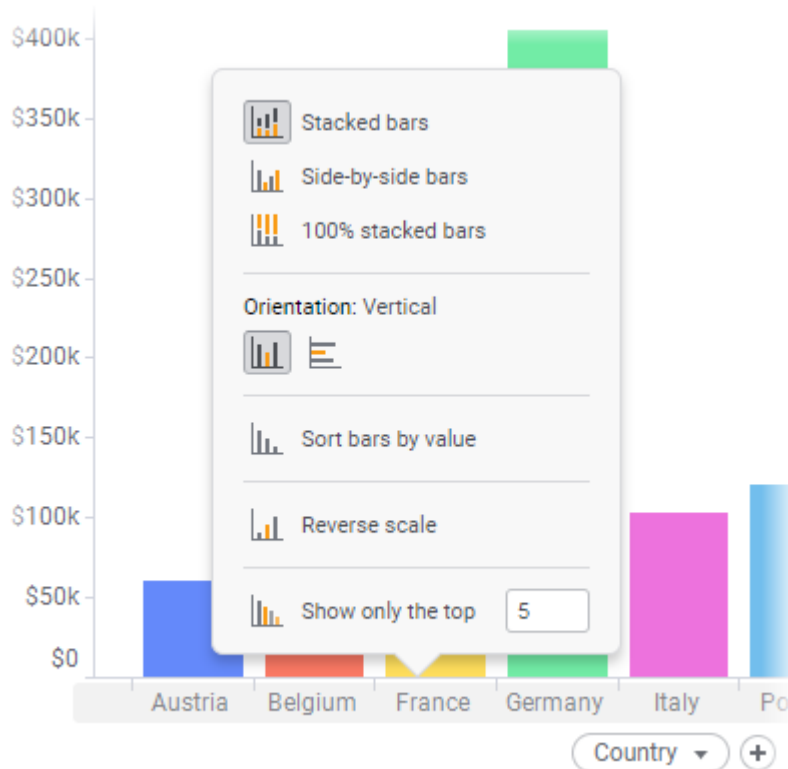
Sometimes only the highest bars are of interest in a bar chart. In these cases, you can select how many of the highest bars to display and let the other bars be hidden.

Assume for example that only the top three bars in the bar chart below are of importance, and the other bars are not relevant to display. Then you can specify that only these three bars should be visible in the visualization.



## Procedure

1. Place the cursor on the **Category** axis in the bar chart. The axis is highlighted.
2. Click the highlighted area to access a pop-up menu.



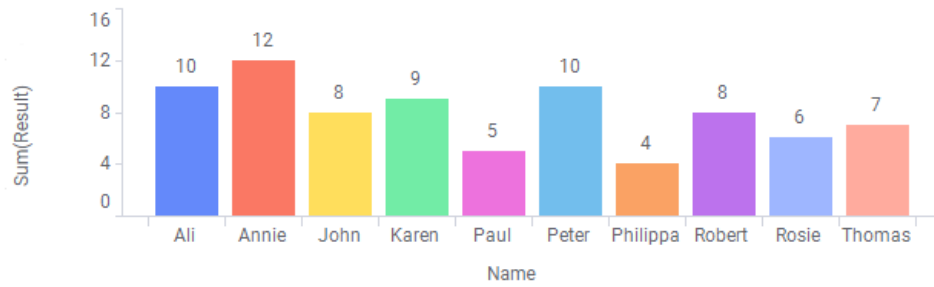
3. In the **Show only the top** field, type how many bars to display, and press Enter.



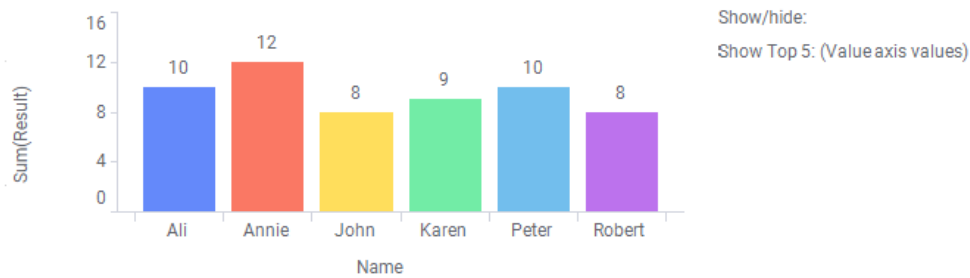
If you want to return to the original bar chart, repeat step 1 and 2, and then simply click **Show only the top** (do not click the text field).

### Example

In this bar chart that shows test results for a number of individuals, more than one bar show the same value.



If 5 is specified in the **Show only the top** field, the resulting bar chart will display six bars as shown below. The reason is that, on the fifth place, there are two bars, John and Robert, with identical values.

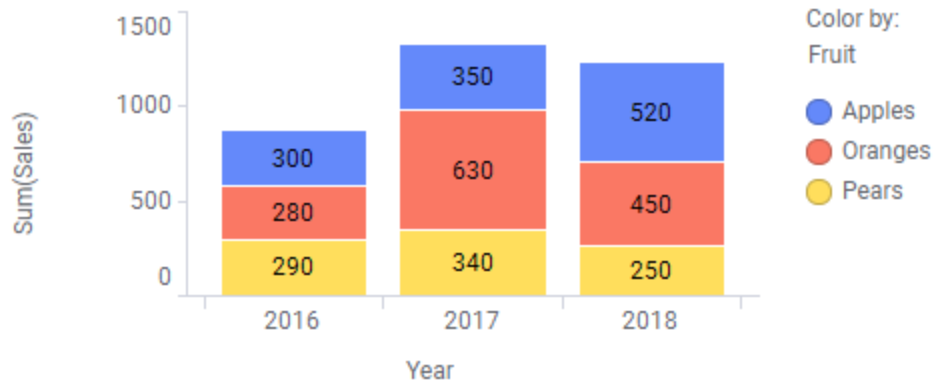


It is possible to show a notification in the legend that only top bars are displayed, as shown above.

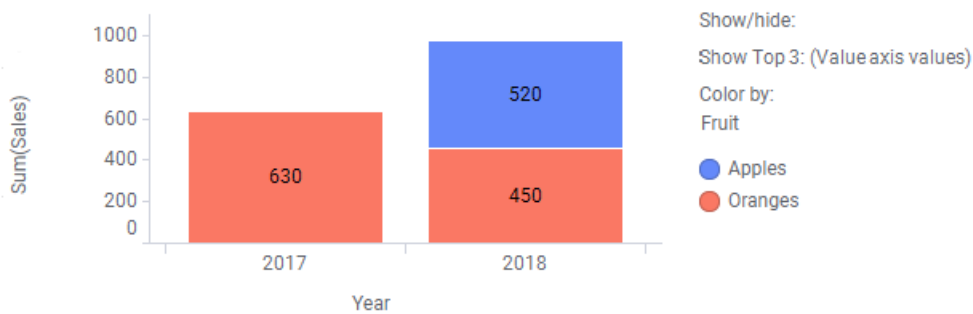
### Showing only top bar segments

If the bars in the bar chart are split into segments, the **Show only the top** setting is applied to the segments and not the entire bars.

In the stacked bar chart below, the yearly sums of different fruit sales are displayed.



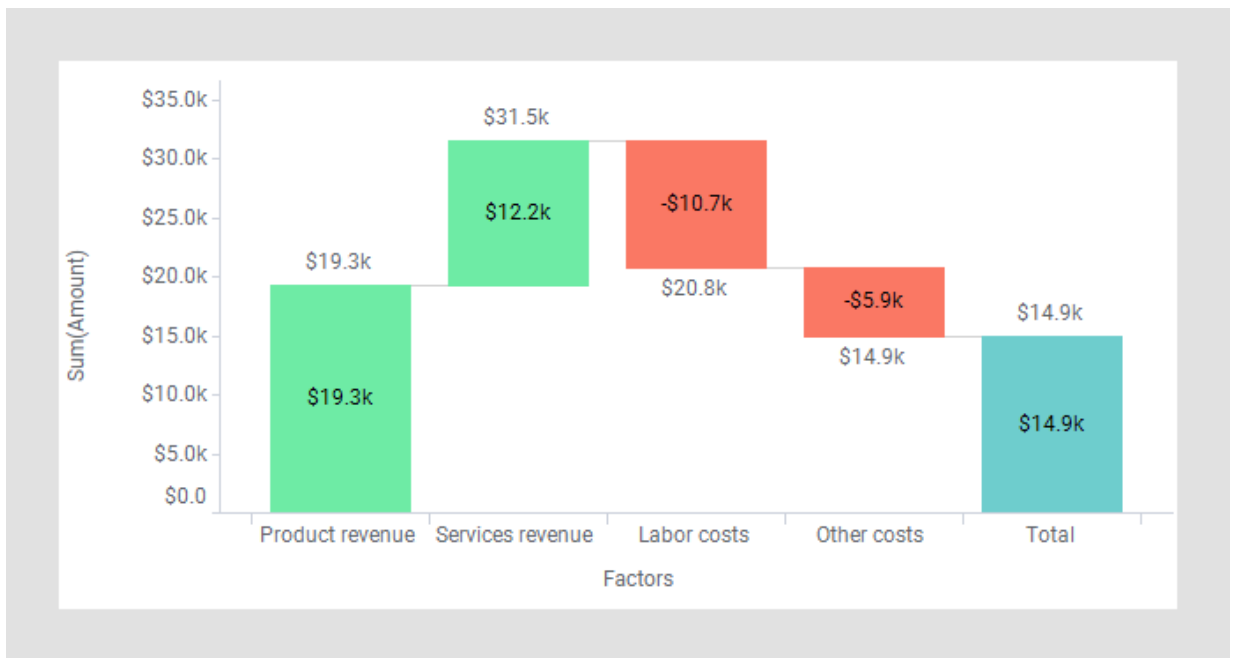
If **Show only the top** is set to 3 in this stacked bar chart, only the top three segments are shown.



## Waterfall chart

A waterfall chart shows how a value changes after being affected by various factors that either increase the value, or decrease it. The resulting value is then presented.

The waterfall chart can, for example, be useful for visualizing the development of a value over time or visualizing the contributions of different factors to a total.



The starting value and the resulting value in the visualization are represented by bars, and the value changes in-between by floating blocks that indicate the ups and downs. As a means to follow the development of the value from start to end, transition lines can be added between the different blocks.

By default, positive value changes are indicated by a green color, and negative value changes by a red color. If you want, you can specify another coloring.

If applicable, bars representing [intermediate calculated totals](#) can be shown.

## Creating a waterfall chart

A waterfall chart is created in slightly different ways depending on how your data is organized, in a tall/skinny or short/wide format.

### Prerequisites

See [Waterfall chart](#) for a description of the concept.

To show the difference between the two formats, the data tables below are used. Both data tables contain the same statistics for population changes over time but are organized differently. In the tall/skinny data table to the left, numbers are provided in a single data column, and in the short/wide data table to the right, the numbers are split on multiple data columns.




Month	Category	Number
Jan	Born	13
Jan	Deceased	-12
Jan	Moved in	17
Jan	Moved out	-5
Feb	Born	10
Feb	Deceased	-8
Feb	Moved in	3
Feb	Moved out	-6
March	Born	6
March	Deceased	-1
March	Moved in	13
March	Moved out	-3

Procedures how to create waterfall charts based on values in a single data column, and values split on multiple columns, follow.

### Waterfall chart based on a single column (tall/skinny data)

#### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Waterfall chart** visualization type to the wanted position on the analysis page. A suggestion of a waterfall chart is presented.
3. On the **Value** axis, [select the column](#) containing the values to be visualized.
4. [Select which type of aggregated value](#) to display for the selected column.



Because the waterfall chart in its nature adds up values, make sure the aggregation type is meaningful. Typically Sum is selected as the aggregation type.

5. On the **Category** axis, select the column containing the factors over which the value changes.

#### Result

The waterfall chart is drawn representing positive value blocks in green, and negative value blocks in red.



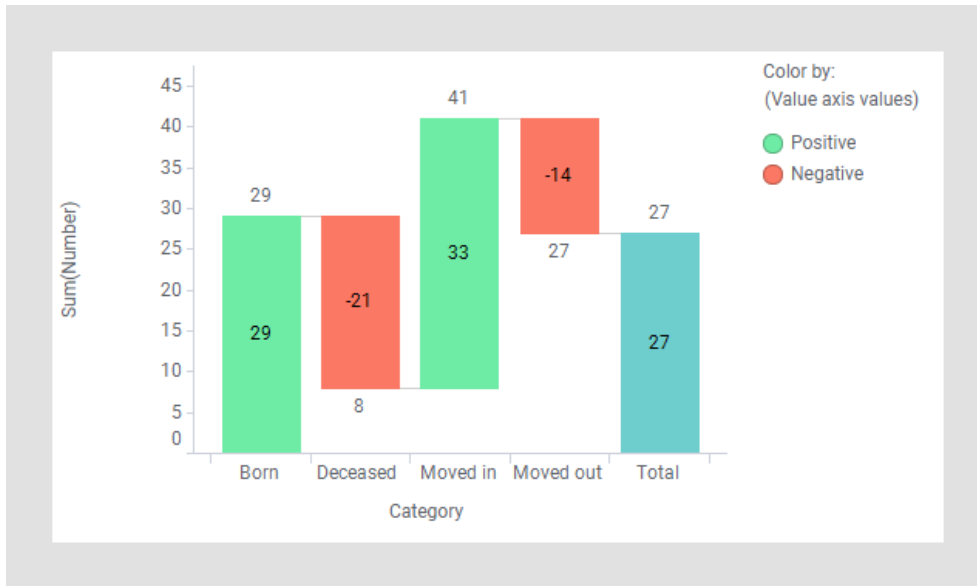
The predefined coloring in green and red can be changed using the **Color by** selector.



You can specify another order in which the factors are drawn. For more information, see [Changing the default sort order of values in a column](#).


### Example

The waterfall chart below is based on the data table above with numbers in a single column. It gives an overview of the population changes for the entire time period.



### Waterfall chart based on multiple columns (short/wide data)

#### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Waterfall chart** visualization type to the wanted position on the analysis page. A suggestion of a waterfall chart is presented.
3. On the **Value** axis, [select the columns](#) containing the values to be visualized.



When the number of columns added to an axis exceeds three, the name of each column is no longer displayed, only how many columns are selected.

4. For each of the selected columns on the **Value** axis, [select which type of aggregated value](#) to display.



Because the waterfall chart in its nature adds up values, make sure the aggregation type is meaningful. Typically Sum is selected as the aggregation type.

5. On the **Category** axis, make sure (Column Names) is selected.

#### Result

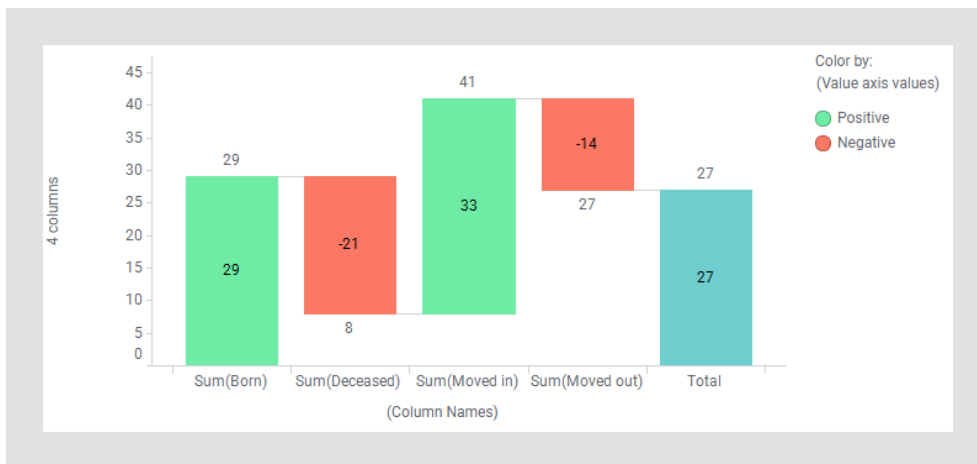
The waterfall chart is drawn representing positive value blocks in green, and negative value blocks in red.



The predefined coloring in green and red can be changed using the **Color by** selector.

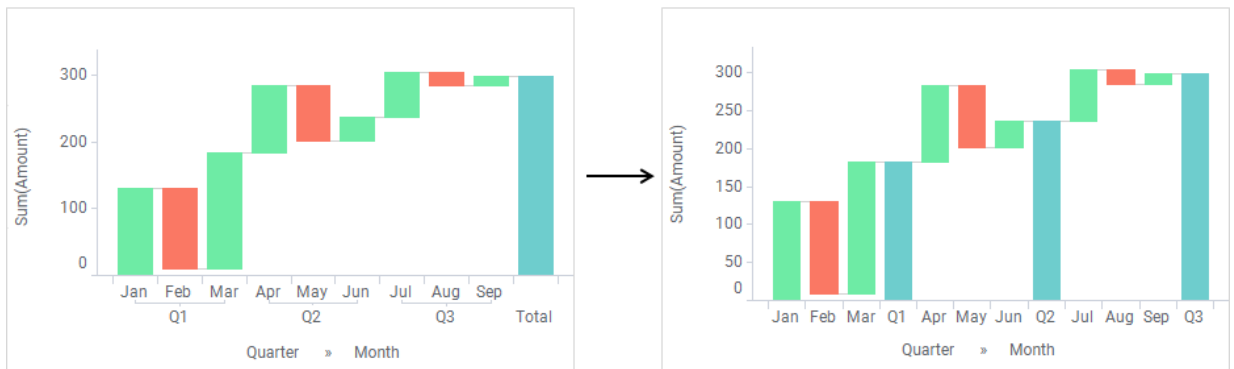
### Example

This waterfall chart is based on the data table above with the numbers provided in multiple columns. It gives the same overview of the population changes as shown above.



### Showing intermediate total bars

In the waterfall chart, bars that show intermediate running totals can be shown. They can be displayed in waterfall charts where a categorical coloring is applied, and in waterfall charts with a hierarchy on the category axis.



### Prerequisites

A hierarchy must be specified on the category axis, or a categorical coloring must be applied on the color axis.

### Procedure

1. Right-click the waterfall chart, and select **Properties** in the opened menu. The Properties pop-over is shown.
2. Click **Total bars**. The **Total bars** section is shown.
3. Make sure the **Show total bars** check box is selected.



You can use the color palette to specify the color of the intermediate total bars.

4. Select the **One total bar per value in** radio button.

5. In the drop-down menu, select the column containing the values after which you want to display totals.

### Result

The intermediate total bars are shown.



Adding intermediate total bars to a hierarchically structured waterfall chart displays the category axis scale labels in a flat structure as shown in the upper-right waterfall chart.

### Hiding total bars

In a waterfall chart, you can encounter total bars showing the resulting value for an entire waterfall or bars representing intermediate totals. You can hide these bars.

#### Procedure

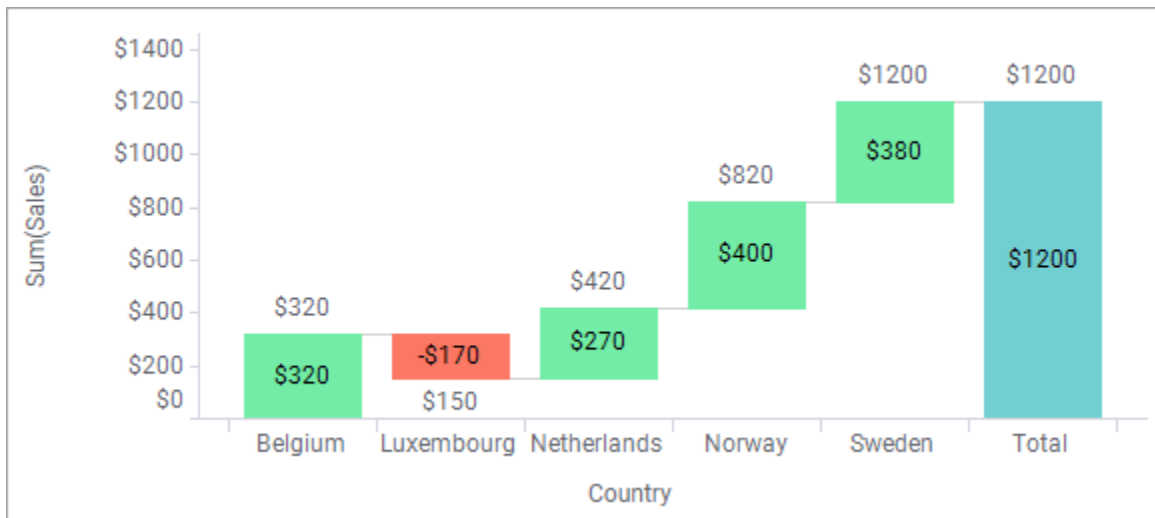
1. Right-click the waterfall chart, and select **Properties** in the opened menu.  
The Properties pop-over is shown.
2. Click **Total bars**.  
The **Total bars** section is shown.
3. Clear the **Show total bars** check box.

### Showing customized intermediate total bars

The intermediate total bars in a waterfall chart usually summarize categories at certain levels in a hierarchically organized data. However, you can customize which categories an intermediate total bar should represent.

The example below illustrates how this can be done using [grouping categories](#).

The waterfall chart shows sales figures for a number of countries. Assume you want to show one intermediate total for the Benelux countries (Belgium, Netherlands, and Luxembourg), and one intermediate total for the Nordic countries (Norway and Sweden), but still be able to view the contribution from each of these countries.



#### Procedure

1. In the waterfall chart, [mark](#) the Belgium, Luxembourg, and Netherlands blocks.

2. Right-click the waterfall chart, and select **Group from marked categories**.  
The Group from marked categories dialog opens, see below.
3. In the drop-down list, select the column containing the values to group.
4. Check that correct axis is selected, and correct values.
5. Type a name of the new group, Benelux, and click **OK**.

Group from marked categories

Which column contains the values to group?

Country

This is currently used on:  
Category axis

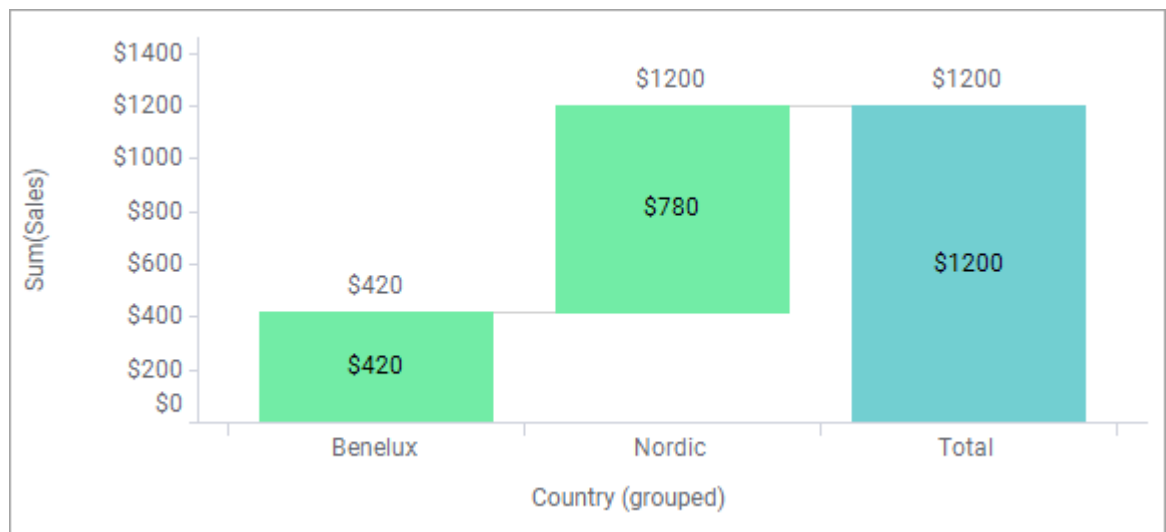
The values you have selected:  
"Belgium", "Luxembourg", "Netherlands"

Give the group a name:

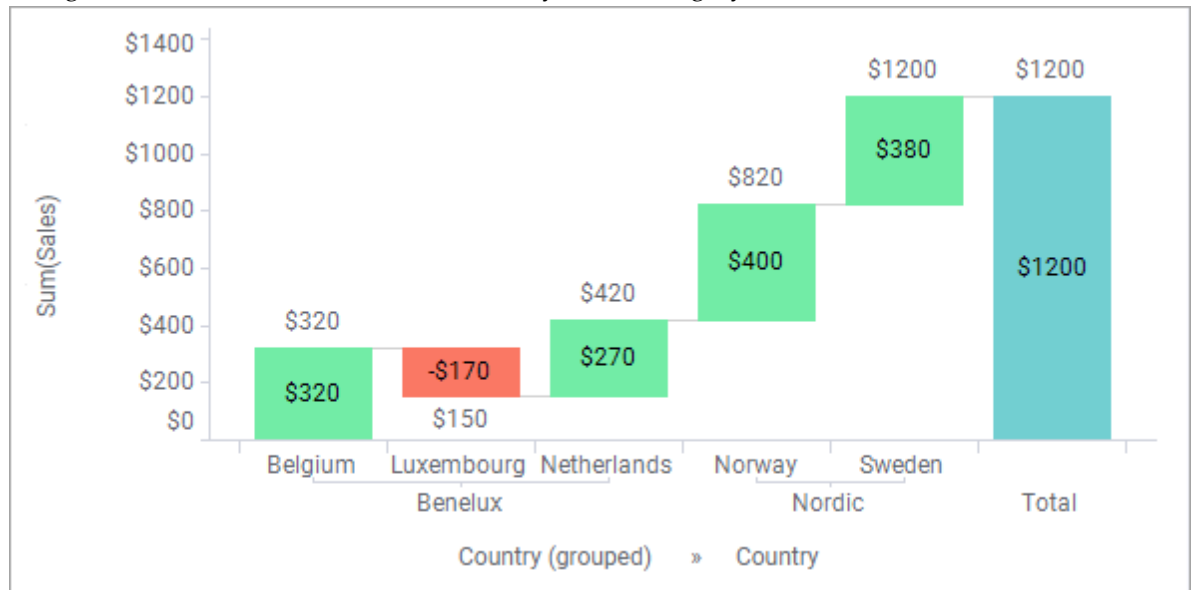
Benelux

OK Cancel

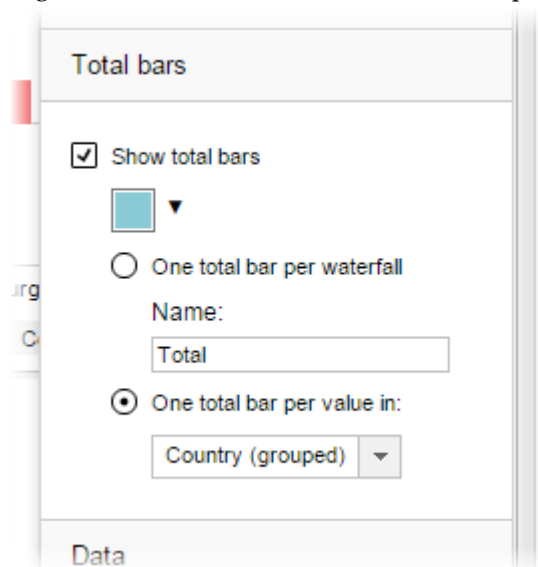
6. In the same way as described above, group Norway and Sweden into a Nordic value.  
A new data column, named Country (grouped) and containing the Benelux and Nordic values, is created.



7. Using the new data column, create a hierarchy on the category axis as shown below.

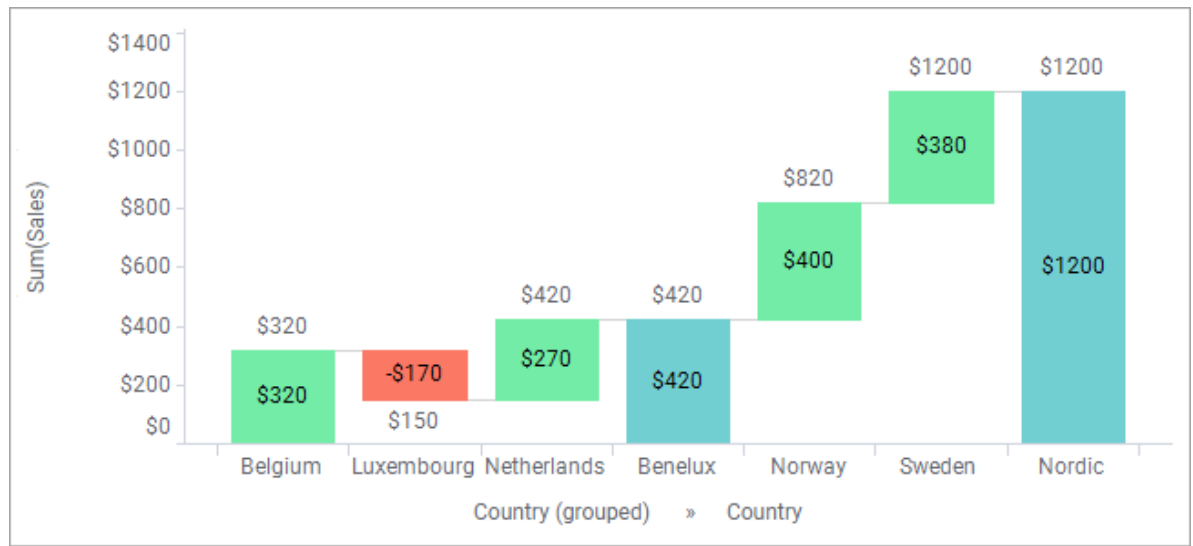


8. Right-click the waterfall chart, and in the opened Properties dialog, click **Total bars**.



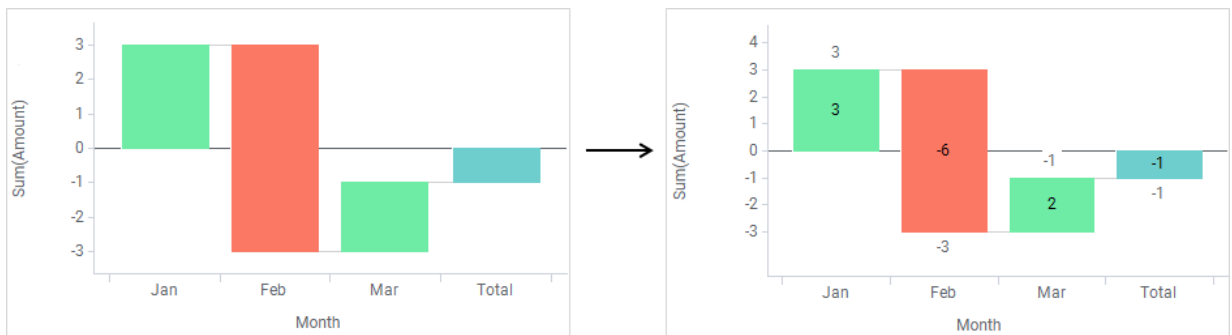
- Select **One total bar per value in**, and make sure Country (grouped) is selected in the drop-down list.

Intermediate total bars for Benelux and Nordic are shown.



## Showing values of bars and blocks

You can show the values of the bars and blocks in the waterfall chart.

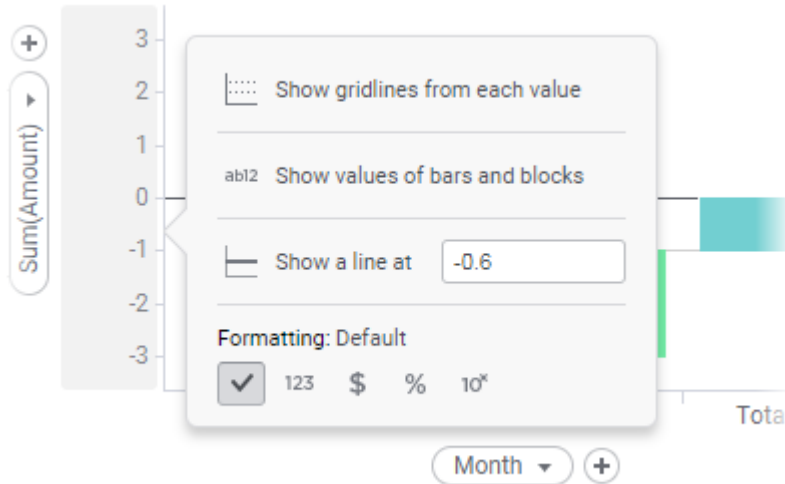


The values of the actual blocks as well as the running totals can be displayed.

### Procedure

- Place the cursor on the value axis.  
The axis is highlighted.

- Click the highlighted area to access a pop-up menu.



- Select **Show values of bars and blocks**.



If you want to remove the values from the bars and blocks, repeat the steps above.



It is possible to display either running total labels or block labels. Open the Properties popover, click **Labels**, and select **Running totals** or **Blocks**, respectively.

## Showing waterfall per categorical color

In a waterfall chart where a categorical coloring is applied, it is possible to show one waterfall per color.

### Prerequisites

A categorical column is selected on the color axis in the waterfall chart.

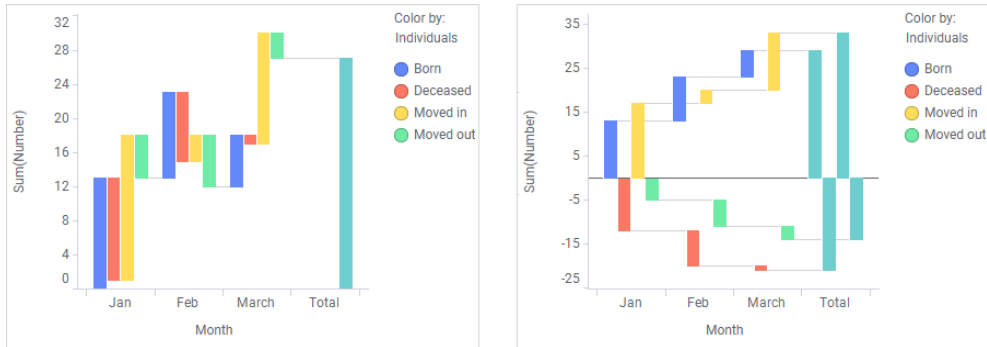
### Procedure

- Right-click the waterfall chart, and select **Properties** in the opened menu.  
The Properties pop-over is shown.
- Click **Layout**.  
The **Layout** section is shown.
- Beneath **Show one waterfall**, select **Per categorical color**.



## Example

The waterfall charts below display population changes over time where each contributing category to the total is represented by a certain color. In the waterfall chart to the left, all categories are factors in the same waterfall. The waterfall chart to the right shows the same data split into separate waterfalls, one per categorical color.



## Showing waterfall per category on the category axis

In a waterfall chart where a categorical coloring is applied, it is possible to show one waterfall per category on the category axis.

### Prerequisites

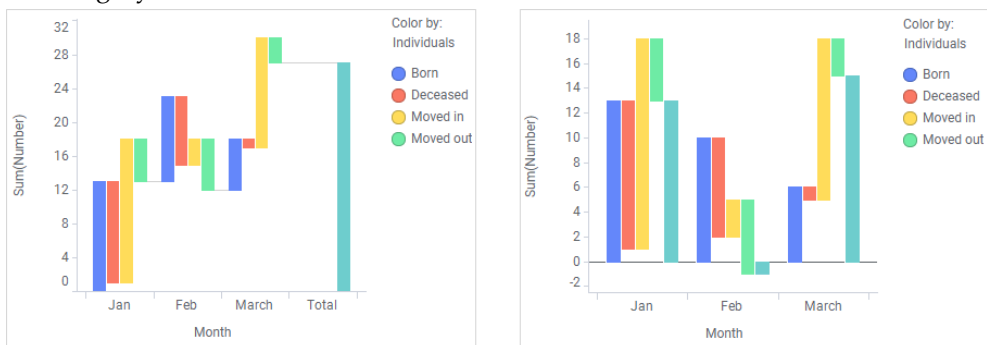
A categorical column is selected on the color axis in the waterfall chart.

### Procedure

1. Right-click the waterfall chart, and select **Properties** in the opened menu. The Properties pop-over is shown.
2. Click **Layout**. The **Layout** section is shown.
3. Beneath **Show one waterfall**, select **Per value on category axis**.

## Example

The waterfall charts below display population changes over time where each contributing category to the total is represented by a certain color. In the waterfall chart to the left, all categories are factors in the same waterfall. The waterfall chart to the right shows the same data split into separate waterfalls, one per category on the category axis.



## Continuing a waterfall over trellis panels

In a trellised waterfall chart, the waterfall by default starts from zero on the value axis in each of the trellis panels. Another option is to let the waterfall continue over multiple trellis panels.

When you continue the waterfall in another [trellis](#) panel, the last value determines where the first value in the next panel should start.

### Prerequisites

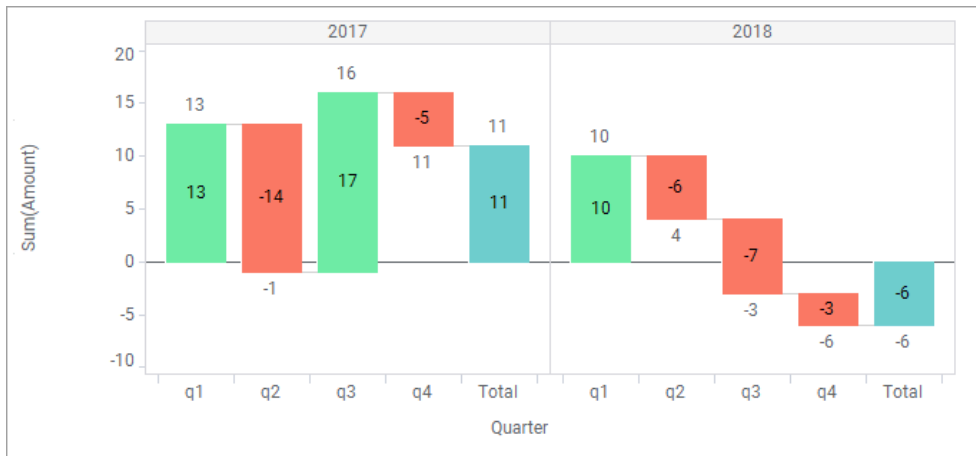
The waterfall chart is trellised.

### Procedure

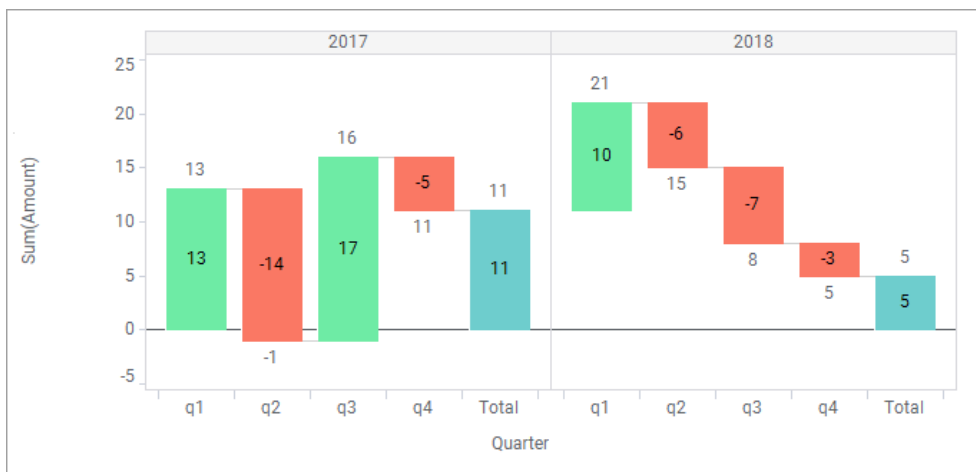
1. Right-click the waterfall chart, and select **Properties** in the opened menu.  
The Properties pop-over is shown.
2. Click **Layout**.  
The **Layout** section is shown.
3. Select the **Continue over trellis** check box.  
You can click the arrow next to the check box to select over which trellis panels the waterfall should continue; **Pages**, **Panels**, **Rows**, and **Columns**.

## Example

The waterfall chart below is trellised into two panels, one panel for 2017 and one panel for 2018. For each of the two panels, the initial value of the waterfall starts at zero. This is the default behavior when trellising a waterfall chart.

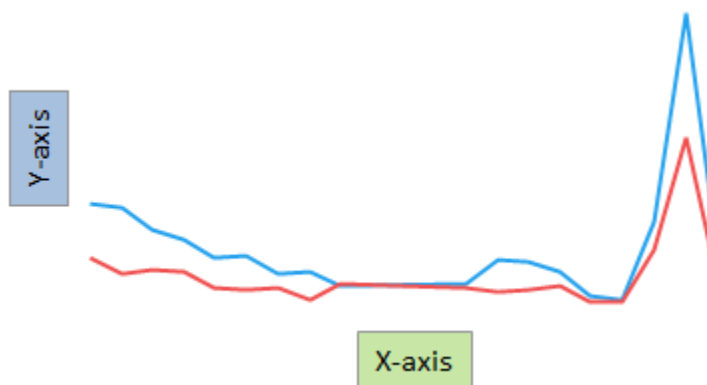


Selecting **Continue over trellis** results in the following waterfall. The first value in the 2017 panel starts where the last value in the 2018 panel ended.



## Line chart

A line chart is used for showing trends, and in most cases trends over time. It can also be used for discerning certain patterns.

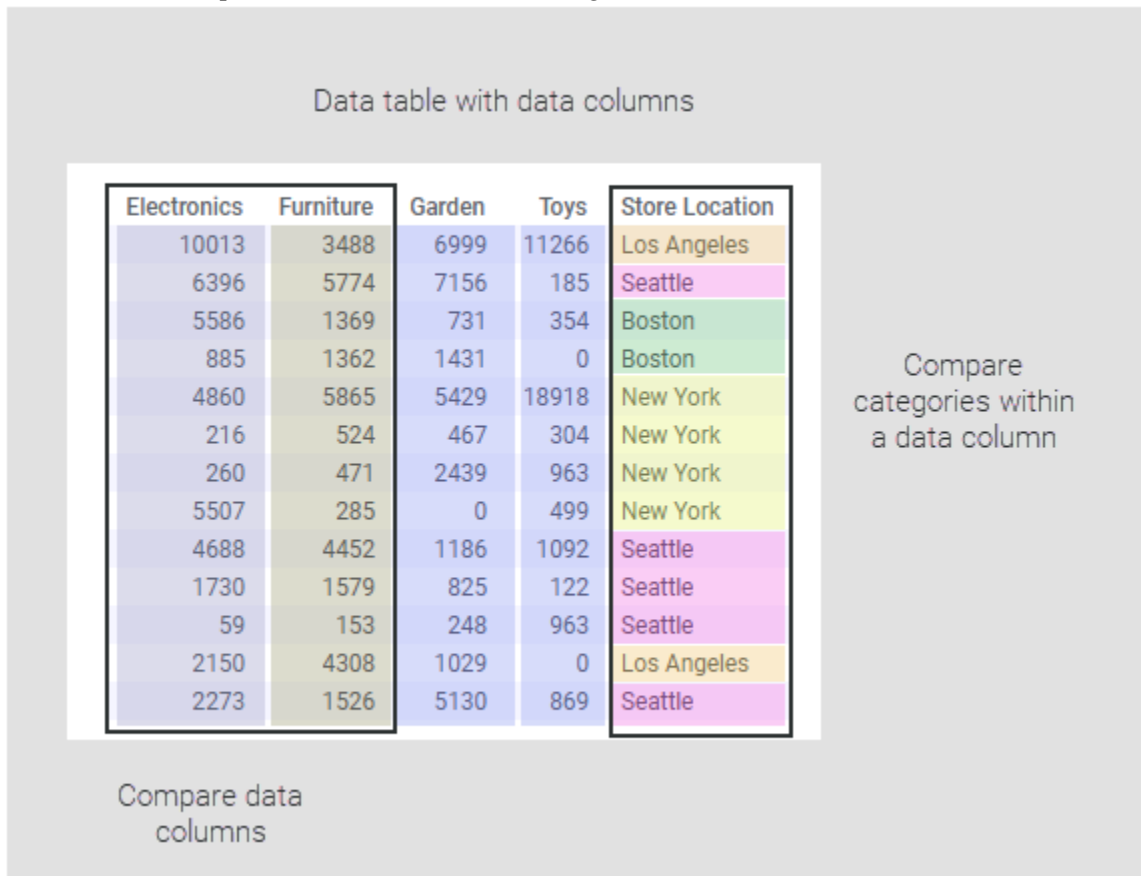


The main axes in the line chart are the X-axis and the Y-axis. To show trends over time, select a time column on the X-axis. For each data point on the X-axis, Y-axis values are placed. These values can represent aggregated data or non-aggregated data for the particular data point. An aggregated value could be, for example, a sum or an average. To draw the "full" line, straight lines are then drawn between these values.

## Creating a line chart


A line chart is used for showing trends, and in most cases trends over time. It can also be used for discerning certain patterns. Depending on what you want to show, or on how your loaded data is organized, you set up the line chart in different ways.

The line chart can consist of one line or of several lines. Several lines are added if you want to make comparisons of several data columns, or if you want to make comparisons between different categories within a column.



## Comparing data columns

### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Line chart** visualization type to the wanted position on the analysis page. A suggestion of a line chart is presented.
3. On the X-axis, [select a time column or any other column](#) for which values should be compared.

Values in time columns get a chronological order automatically. To display time data hierarchically, see [Specifying time hierarchies](#).

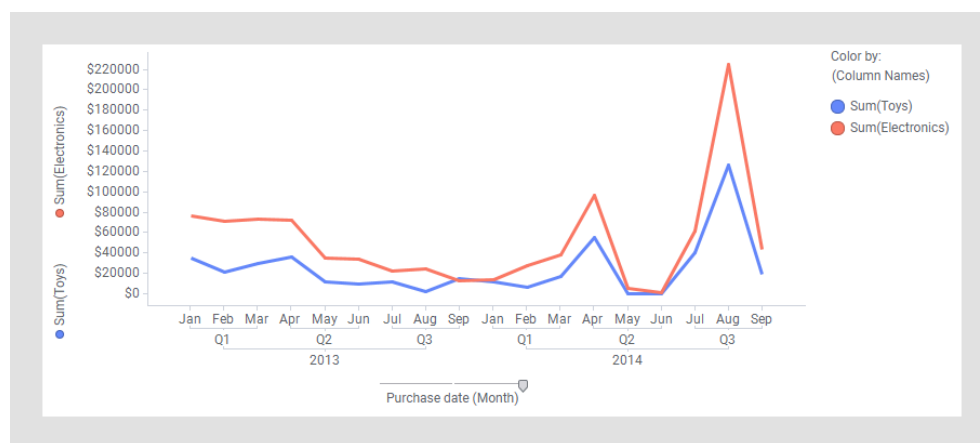
- On the Y-axis, [select the columns](#) you want to present as separate lines.  
One line per column is drawn using different colors. The **Color** axis in the legend is set to (Column Names).
- For each of these columns, [select which type of aggregated value](#) to display.

### Example

Assume the data below is loaded. It contains customers' electronics and toys purchases made at different store locations during a time period.


Customer ID	Store Location	Date Joined	Electronics	Toys
SSMM55001	Los Angeles	7/21/2014	10013	11266
SSMM55002	Seattle	7/23/2014	6396	185
SSMM55003	Boston	7/23/2014	5586	354
SSMM55004	Boston	7/23/2014	885	0
SSMM55005	New York	7/23/2014	4860	18918
SSMM55006	New York	7/23/2014	216	304
SSMM55007	Los Angeles	7/23/2014	436	0
SSMM55008	New York	7/23/2014	0	0
SSMM55009	Los Angeles	7/23/2014	153	0
SSMM55010	Boston	7/24/2014	609	1934
SSMM55011	Boston	7/25/2014	1384	91
SSMM55012	New York	7/25/2014	260	963
SSMM55013	New York	7/26/2014	5507	499
SSMM55014	New York	7/26/2014	938	0
SSMM55015	Seattle	7/26/2014	1588	1000

Below see the settings made to compare the sums of purchases made at the Electronics and Toys departments.



## Comparing categories within a column

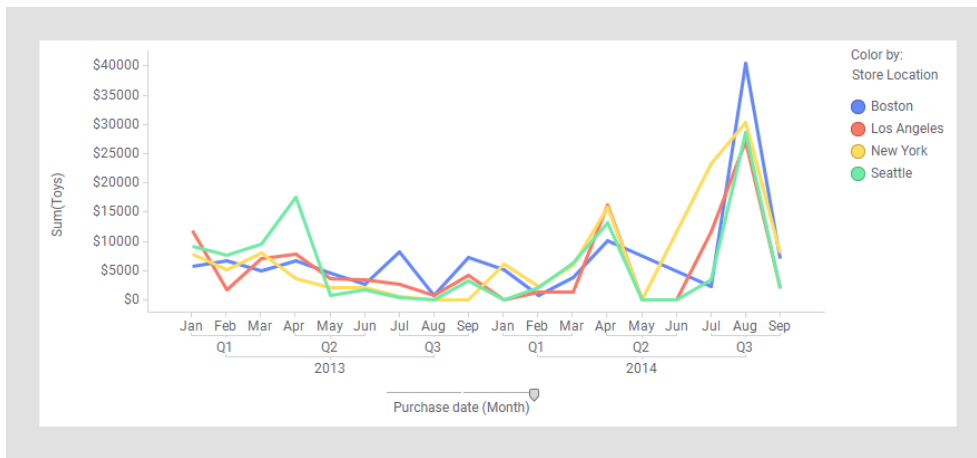
### Procedure

- On the authoring bar, click **Visualization types**  to open the flyout.

2. Drag the **Line chart** visualization type to the wanted position on the analysis page.  
A suggestion of a line chart is presented.
3. On the X-axis, [select a time column or any other column](#) for which values should be compared.  
Values in a time column get a chronological order automatically.
4. On the Y-axis, select the column whose values you want to present.
5. [Select which type of aggregated value](#) to display for the selected column.
6. On the **Color** axis in the legend, specify the column containing the categories you want to present as separate lines.

### Example

The data shown in the example above is loaded. Below see the settings made to compare the sum of Toys purchases at different store locations.



### Marking in line charts

You can mark entire lines and parts of lines.

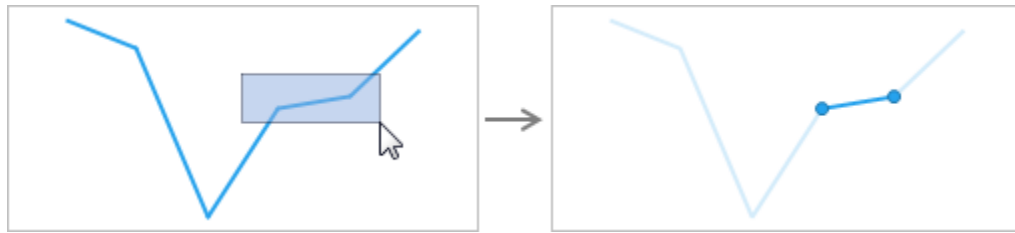
By clicking a line anywhere, the entire line is marked.

If you want to mark only parts of a line, drag a rectangle, or press Alt and draw a shape with the mouse over the part in question. Bear in mind that the lines in a line chart consist of data points, which are connected by straight lines.

- If one data point is included, that point is marked.



- If data points next to each other on the line are included, the points and the line between the points are marked.

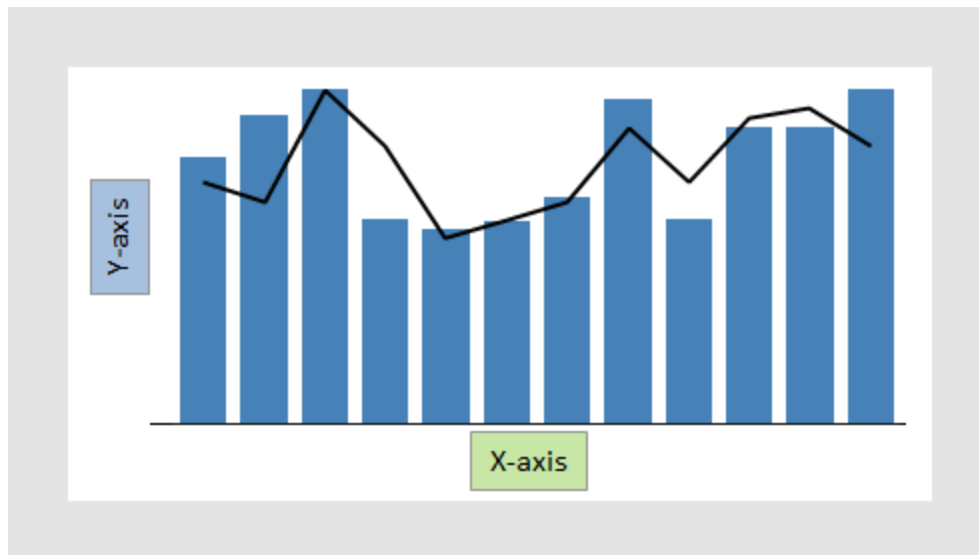


- Data points in between that are not included are not marked.



## Combination chart

The combination chart is a visualization that combines the features of the bar chart and the line chart.



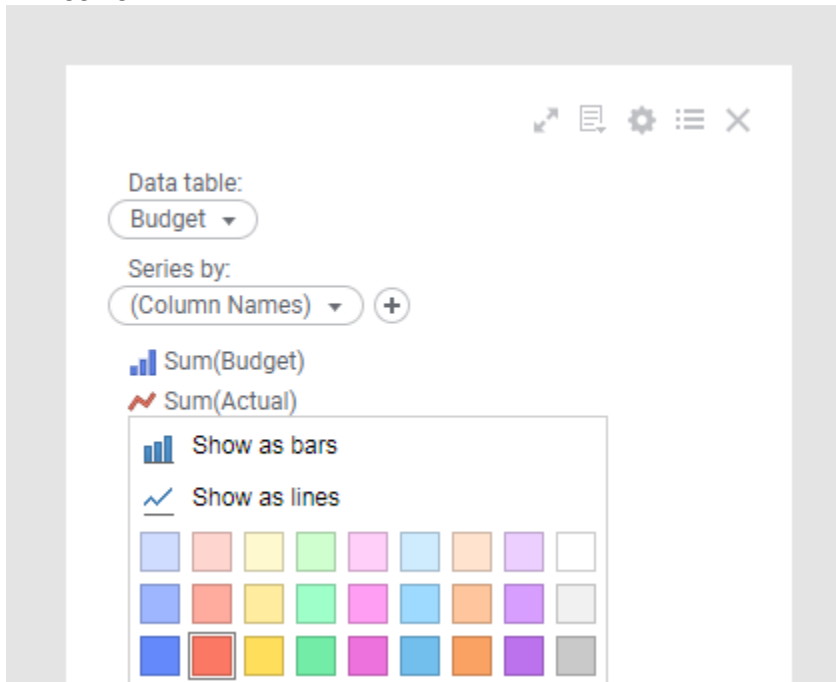
In a combination chart, you can display both bars and lines in a single visualization. Because of the overlay effect, where lines are drawn on top of the bars, it is easy to compare values for different columns or categories in your data. Trends can be identified, and you can spot deviations directly. For example, you can use the combination chart to compare the projected sales with the actual sales for different time periods.

The combination chart has similarities to both bar charts and line charts. As with these visualizations, columns containing numerical data are usually selected on the Y-axis. For each X-axis data point, the data is aggregated into values on the Y-axis. An aggregated value could be, for example, a sum or an average. These values are then displayed either in the form of bars or in the form of a line for each column or category to be compared.

## Series by

Similarly to the function of **Color by** in other visualizations, **Series by** in the combination chart is a way to divide the data into slices. The difference is that the slices in the combination chart, called series, can be defined as bars, or lines, as well as being colored separately. That is, each series in the combination chart will be represented by a line or a set of bars in the visualization.

You can separate the different columns or categories to be compared by coloring them differently. The colors are specified on the **Series** axis in the legend, and this is also where you specify whether to show the aggregated values as bars or lines.



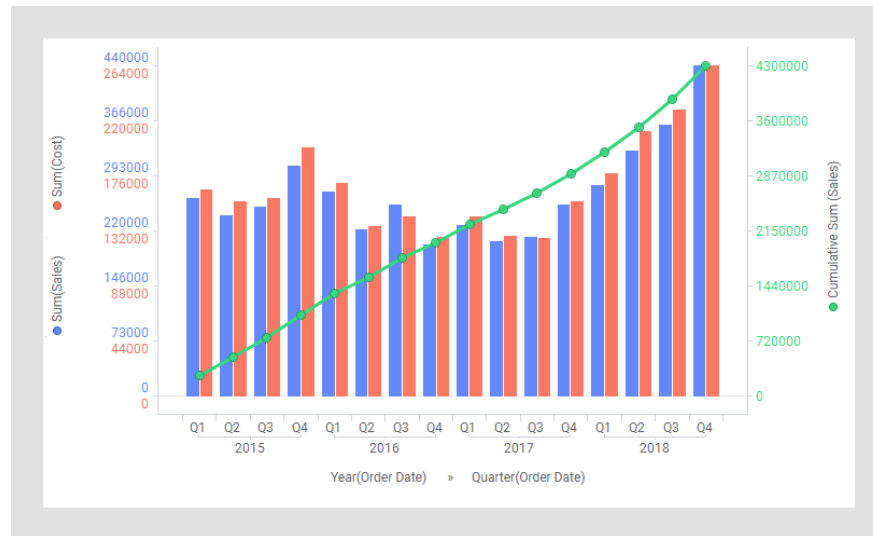
On the X-axis, columns containing categorical values are selected, or columns with time data when comparing data over time.

As mentioned above, you might want to compare columns in your data, or you might want to compare different categories within a column, as illustrated below.



### Example: Multiple scales on the Y-axis

With dual or multiple scales on the Y-axis, you can compare several lines and bars with significantly different value ranges. See [Adjusting scales on axes](#).



### Creating a combination chart

In a combination chart, you have the option to display both bars and lines in a single visualization. Because of the overlay effect, lines are drawn on top of the bars, it is easy to compare values for different columns or categories in your data. Trends can be identified, and you can spot deviations directly.

Depending on what you want to show, or on how your loaded data is organized, you set up the combination chart in different ways.

Data table with data columns


Electronics	Furniture	Garden	Toys	Store Location
10013	3488	6999	11266	Los Angeles
6396	5774	7156	185	Seattle
5586	1369	731	354	Boston
885	1362	1431	0	Boston
4860	5865	5429	18918	New York
216	524	467	304	New York
260	471	2439	963	New York
5507	285	0	499	New York
4688	4452	1186	1092	Seattle
1730	1579	825	122	Seattle
59	153	248	963	Seattle
2150	4308	1029	0	Los Angeles
2273	1526	5130	869	Seattle

Compare  
categories within  
a data column

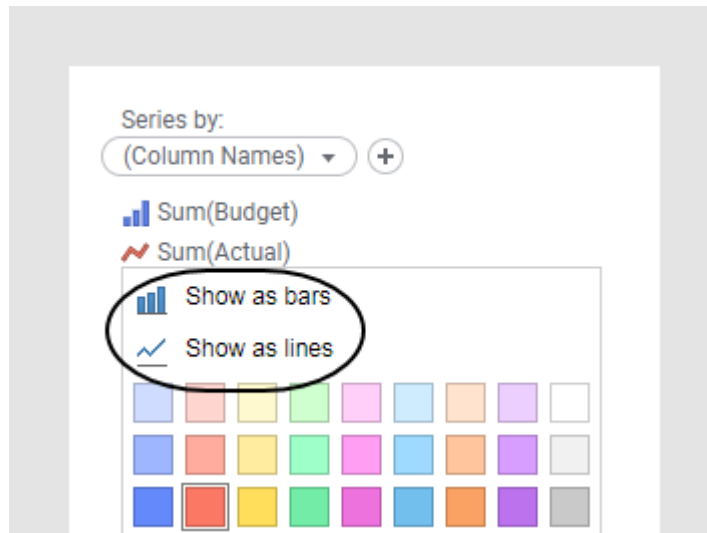
Compare data  
columns

## Comparing data columns

### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Combination chart** visualization type to the wanted position on the analysis page. A suggestion of a combination chart is presented. However, no lines are drawn, only bars.
3. On the Y-axis, [select the columns](#) to present as bars or lines.  
For each selected column, bars are drawn using a certain color. The **Series** axis in the legend is set to (Column Names).
4. For each of these columns on the Y-axis, [select which type of aggregated value](#) to display.

5. On the **Series** axis in the legend, click a column and select display option; **Show as bars** or **Show as lines**.



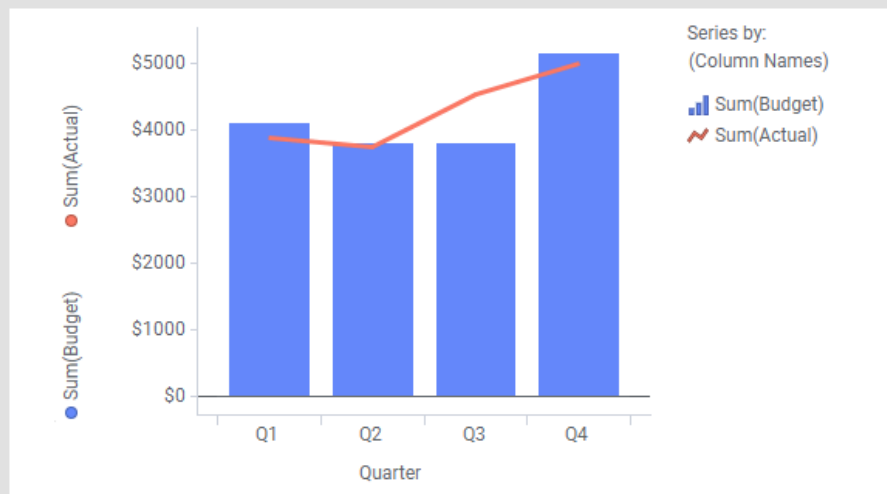
6. On the X-axis, [select one or more columns](#) for which values should be compared.  
If a time column is selected, it can be beneficial to [display the time data hierarchically](#).

### Example

Assume you want to compare on a quarterly basis estimated figures in a budget to actual figures. The "Actual" and "Budget" columns below contain the monthly actual and estimated figures respectively.


Quarter	Month	Budget	Actual
Q1	Jan	1000	1050
Q1	Feb	1500	1100
Q1	Mar	1600	1750
Q2	Apr	1300	1430
Q2	May	1300	1100
Q2	Jun	1200	1220
Q3	Jul	1100	1090
Q3	Aug	1200	1660
Q3	Sep	1500	1800
Q4	Oct	1700	1600
Q4	Nov	1650	1600
Q4	Dec	1800	1800

See the settings made to compare the Actual to estimated Budget figures. Deviations are easily identified, for example, the budget was exceeded a lot in Q3.



### Comparing categories within a column

#### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Combination chart** visualization type to the wanted position on the analysis page. A suggestion of a combination chart is presented. However, no lines are drawn, only bars.

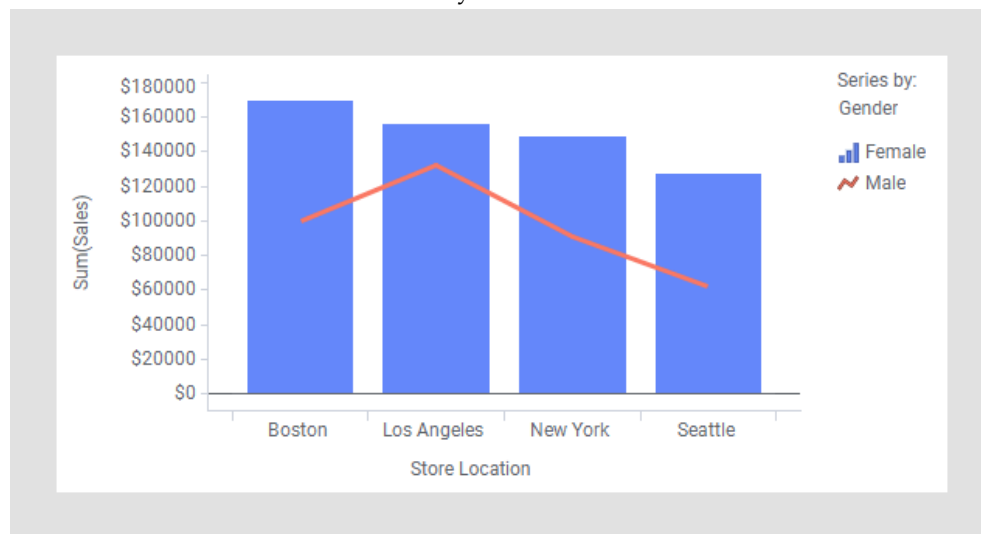
3. On the Y-axis, [select the column](#) whose values you want to present.
4. [Select which type of aggregated value](#) to display for the selected column.
5. On the **Series** axis in the legend, [select the column](#) containing the categories you wish to compare .
6. Then click a category and select display option; **Show as bars** or **Show as lines**.

### Example

The data table contains sales, split per gender, at different store locations.

Store Location	Gender	Sales
Los Angeles	Female	6999
Seattle	Female	7156
Boston	Female	731
Boston	Male	1431
New York	Female	5429
New York	Female	467
Los Angeles	Female	467
New York	Female	0
Los Angeles	Male	0
Boston	Female	747
Boston	Female	405
New York	Male	2439

The following combination chart shows the settings made to display the differences in the sum of sales made by women and men.

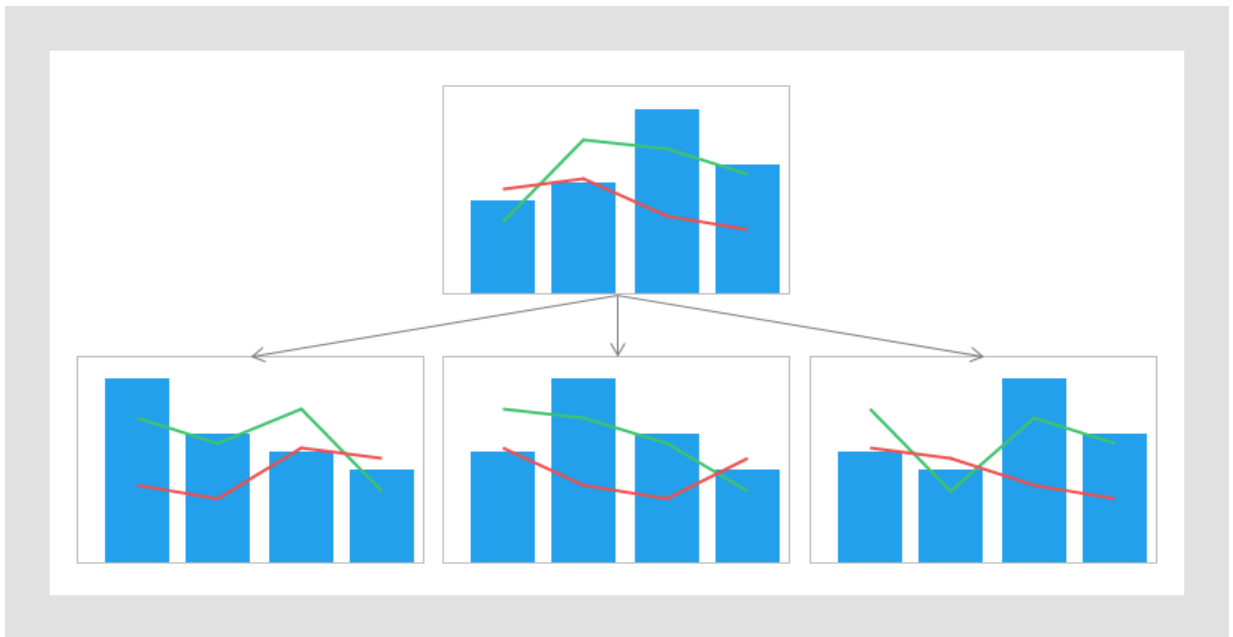


### Sorting the combination chart

You can select a series to sort the X-axis by in a combination chart.

When you sort, the bars or the line in a series are sorted from the highest to the lowest value.

In the image, there is a combination chart with three series; one bar series and two line series. The three combination charts at the bottom show the results when sorting it by respective series.



### Procedure

1. Right-click the combination chart, and select **Properties** in the opened menu. The Properties popover is shown.
2. Click **Layout and sorting**. The **Layout and sorting** section is shown.
3. In the **Sort X-axis by** drop-down menu, select the series to sort by.



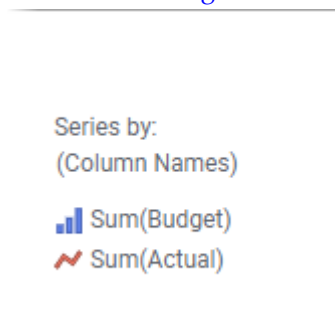
If you want to sort a series from the lowest to the highest value, place the cursor on the X-axis, and click the area that is highlighted. Then select **Reverse scale** in the opened pop-up menu.

## Switching between bars and line series

In a combination chart, a series can be displayed by means of bars or a line. You can switch between these two display options.

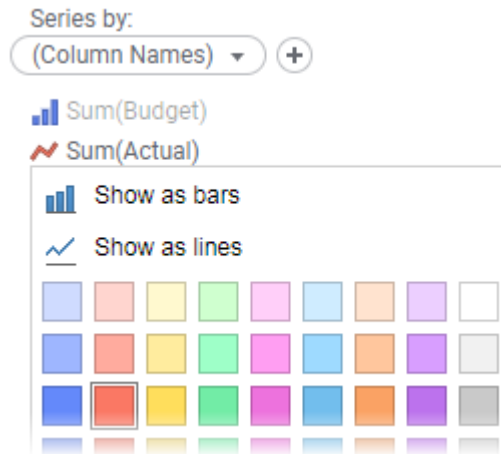
### Procedure

1. Make sure the [legend is visible](#).



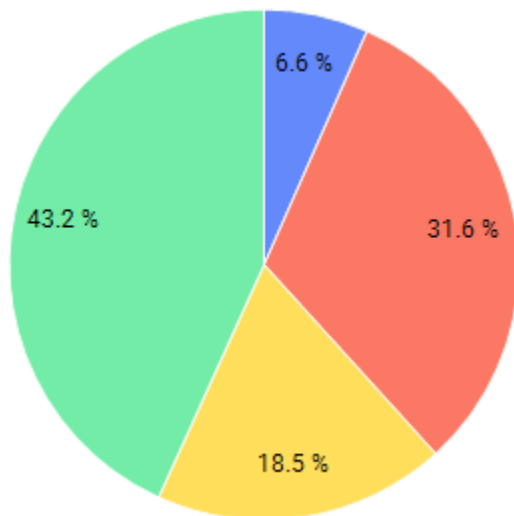
2. Beneath **Series by**, click the symbol for the series you want to change.

3. On top of the opened palette, select the display option; **Show as bars** or **Show as lines**.

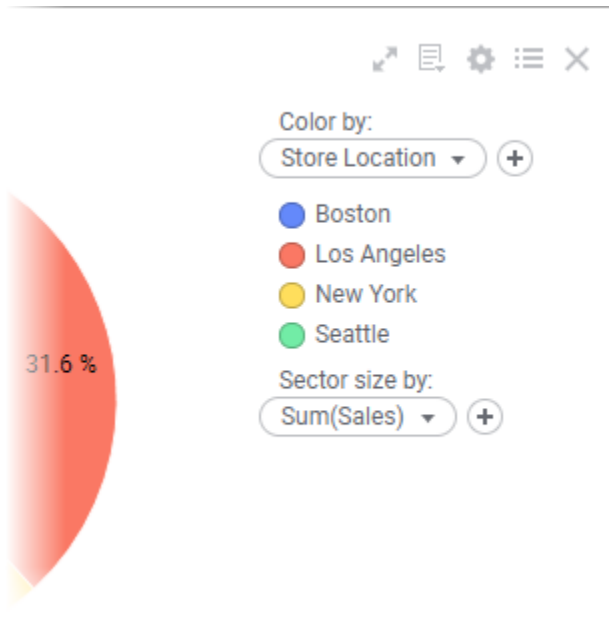


## Pie chart

A pie chart is a circle graph that is divided into sectors. It is used to compare values for different categories in your data on a relative basis. Each pie sector represents a specific category, and its size the category's contribution to the whole value, expressed as a percentage. The values are usually sums.



The main axes in the pie chart are the **Color** axis and the **Sector size** axis in the legend.




You use the **Color** axis to divide the pie into sectors, and the **Sector size** axis to specify what data the sectors' relative sizes should reflect.

## Creating a pie chart

A pie chart is a circle graph that is divided into sectors. It is used to compare values for different categories in your data on a relative basis. Each pie sector represents a specific category, and its size the category's contribution to the whole value, expressed as a percentage. The values are usually sums.

### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Pie chart** visualization type to the wanted position on the analysis page.  
A suggestion of a pie chart is presented.
3. On the **Color** axis in the legend, [select the column](#) containing the categories that you want to present as pie sectors.
4. On the **Sector size** axis, [select the column](#) containing the data that the sizes of the sectors should reflect.
5. For the selected column on the **Sector size** axis, [make sure that the aggregation type](#) is set to Sum.  
There are times, though, when Count or UniqueCount are useful aggregation types.

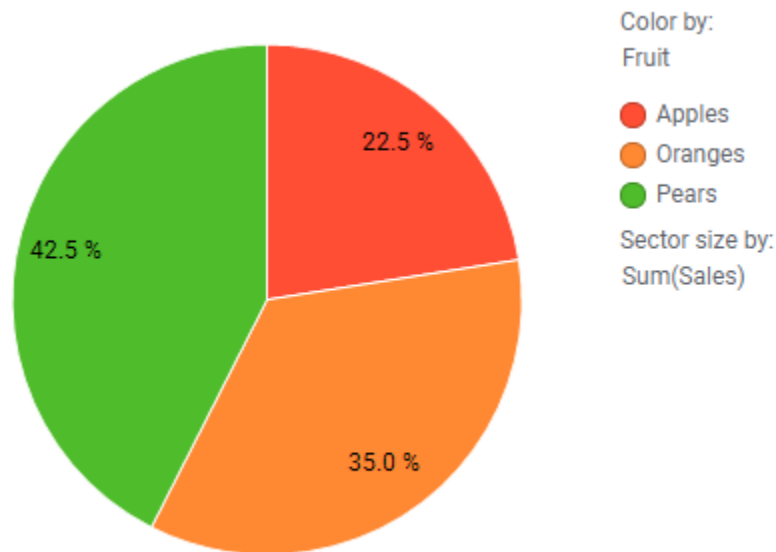


### Example

The data table lists the sales of different fruits.

Fruit	Sales
Apples	20
Pears	30
Oranges	15
Oranges	25
Pears	5
Pears	10
Apples	5
Apples	20
Pears	40
Oranges	30

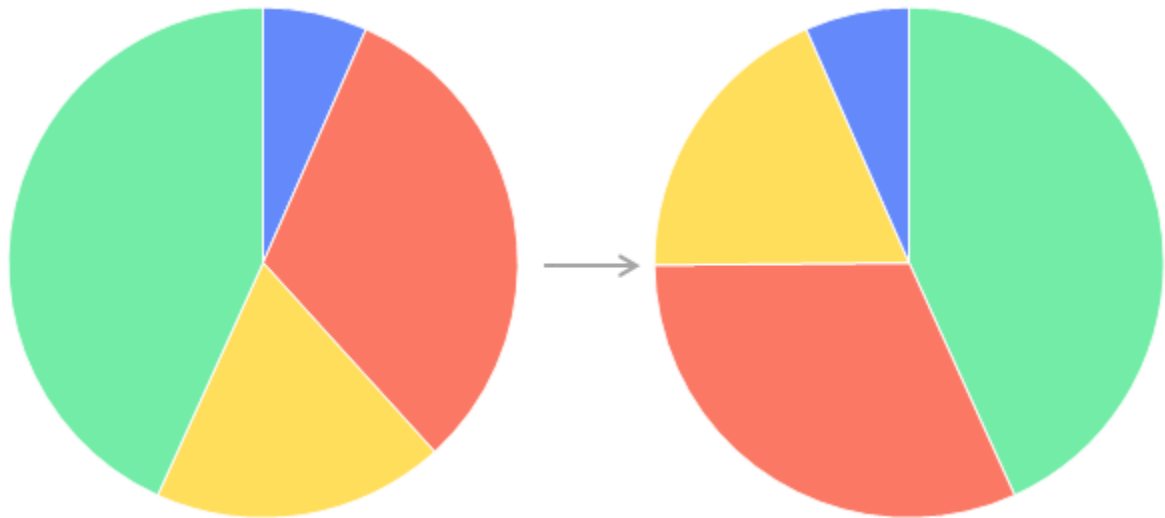
The pie chart below displays the different fruits' contributions to the total sales. Note the settings on the **Color** and **Sector size** axes.



### Sorting pie sectors by size

You can sort the pie sectors within a pie chart.

The pie sectors are sorted by size clockwise with the largest sector at the upper right of the pie chart.

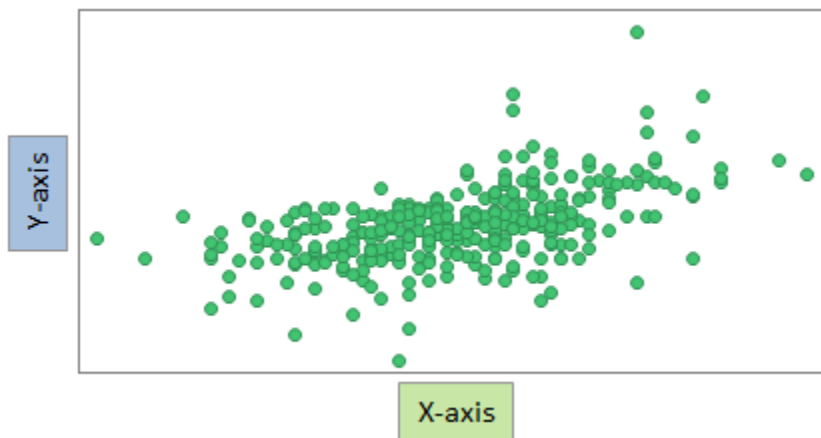


### Procedure

1. Right-click the pie chart, and select **Properties** in the opened menu. The Properties pop-over is shown.
2. Click **Sorting**.
3. Select the **Sort sectors by size** check box.

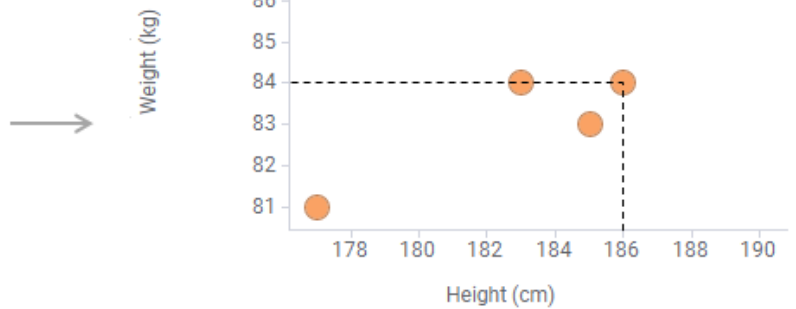
## Scatter plot

In a scatter plot, markers are displayed in a two-dimensional coordinate system. It is useful for getting an overview of how your data is distributed across two dimensions.



By default, each row in your loaded data table is represented by a marker. The marker's position in the coordinate system is defined by the row's values in the columns selected on the X-axis and the Y-axis. For example, the marker's position for the last row in the data table below is the coordinate 186 on the X-axis, and 84 on the Y-axis.


Name	Height (cm)	Weight (kg)
Sean	185	83
Roger	177	81
Timothy	190	88
Pierce	183	84
Daniel	186	84



## Creating a scatter plot

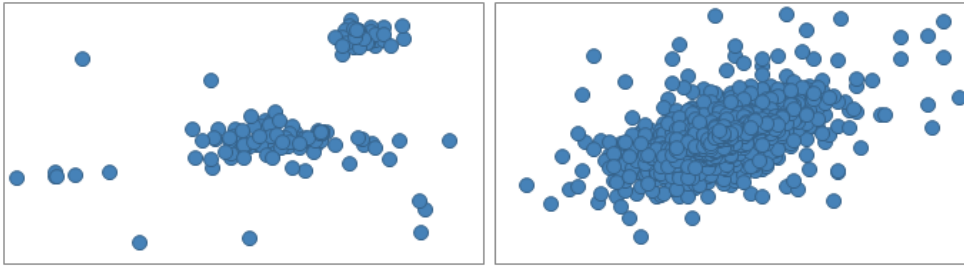
In a scatter plot, markers are displayed in a two-dimensional coordinate system. It is useful for getting an overview of how your data is distributed across two dimensions.

### Procedure

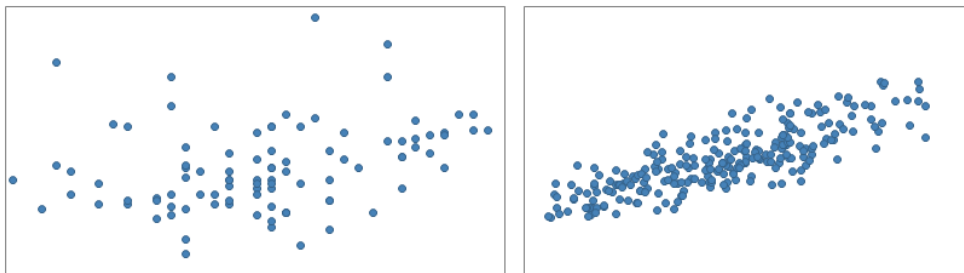
1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Scatter plot** visualization type to the wanted position on the analysis page.  
A suggestion of a scatter plot is presented.
3. [Select the column](#) to display on the X-axis.
4. [Select the column](#) to display on the Y-axis.

## Examples

When visualizing data in scatter plots, you get a view of the distribution. You may for example find that data forms groups, or that data is more gathered.



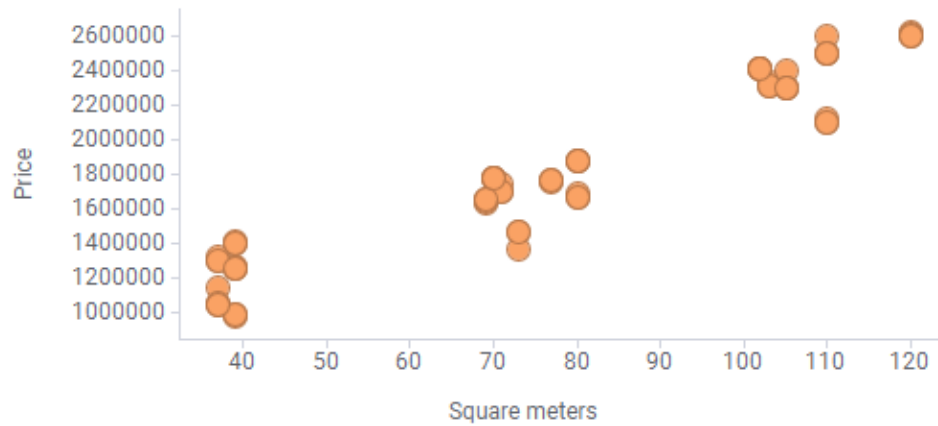
The scatter plot can also be used to examine whether or not there is a relationship between the columns. The closer the markers are to a straight line, the stronger the relation. For example, the column values in the left-hand scatter plot below do not seem to be related, but in the right-hand scatter plot, a higher value on the X-axis seems, in general, to have a higher value on the Y-axis.



To illustrate how you can get an overview of the data distribution, the following data table is visualized in the bottom scatter plot.

Apartment ID	Square meters	Price
1	77	1765000
2	103	2310000
3	110	2100000
4	39	990000
5	73	1465000
6	80	1670000
7	37	1300000
8	71	1700000
9	39	1400000
10	80	1875000
11	120	2600000
12	69	1660000
13	70	1780000
14	37	1050000
15	105	2300000
16	102	2405000

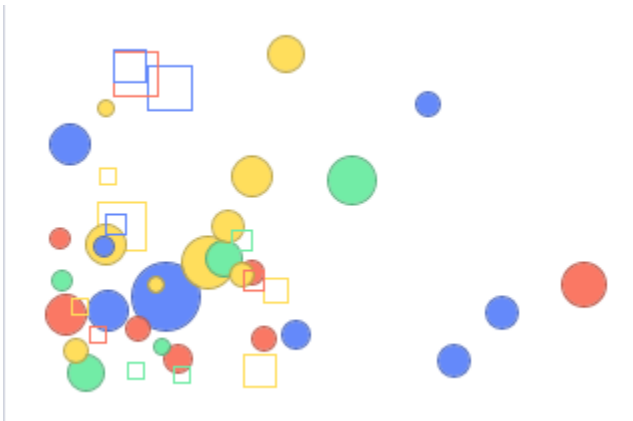
The data table lists areas and prices of a number of apartments. At a quick glance, it is easy to see that the objects appear in groups.



## Using marker appearance to add more information

The markers in a scatter plot provide an overview of how the data is distributed across two dimensions, the X-axis and the Y-axis. If you use colors, shapes, and sizes of the markers, further information about your data can be displayed.

A scatter plot, where all these three marker attributes are used, is shown below.



To make use of these attributes, you simply drag the column you want to visualize from the **Data in analysis** flyout to any of the color, size, or shape drop targets in the scatter plot, alternatively use the scatter plot properties popover.



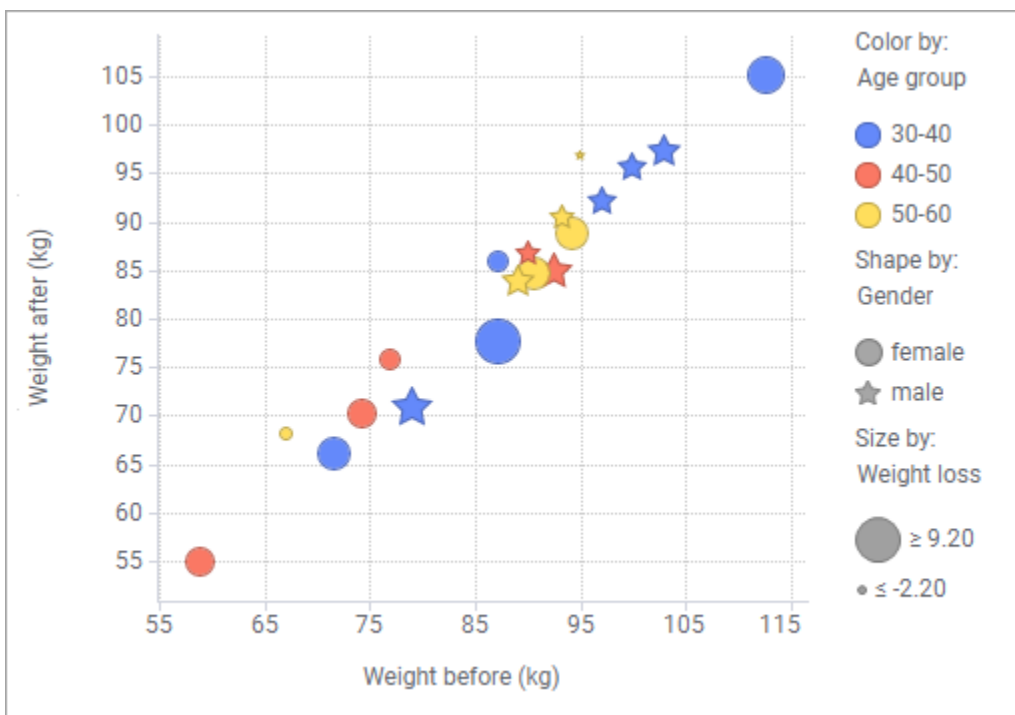
## Example

The data table below contains weights of people, who participate in a weight loss program.

Gender	Weight before (kg)	Weight after (kg)	Age group	Weight loss
female	87.10	85.90	30-40	1.20
female	67.00	68.20	50-60	-1.20
female	74.20	70.30	40-50	3.90
male	93.20	90.50	50-60	2.70
male	92.40	85.00	40-50	7.40
male	102.80	97.30	30-40	5.50
male	99.80	95.70	30-40	4.10
female	71.60	66.20	30-40	5.40
female	90.40	84.80	50-60	5.60
female	76.80	75.80	40-50	1.00
female	58.80	55.00	40-50	3.80
male	89.00	84.00	50-60	5.00

All the data columns are represented in the scatter plot below.

- The positions of the markers on the X- and Y-axis are based on the participants' weights before and after the program.
- The colors distinguish the different age groups.
- The shapes indicate the gender.
- The marker sizes represent the weight losses.

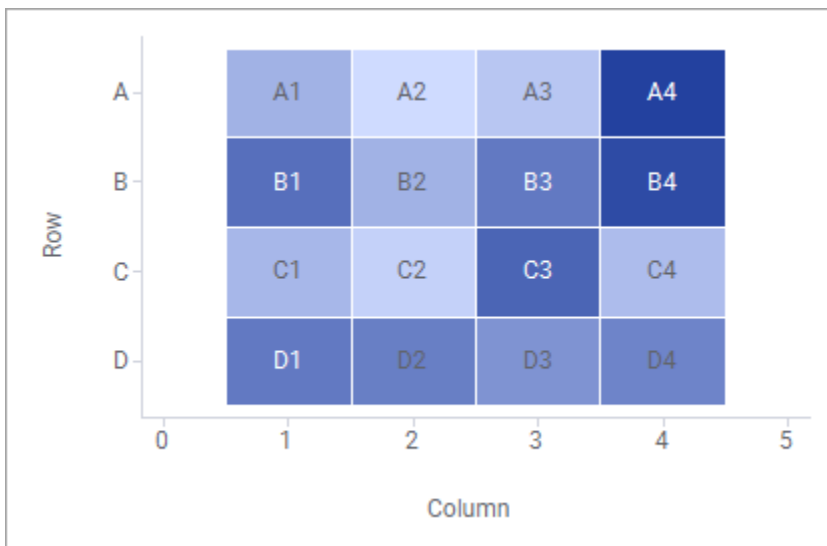
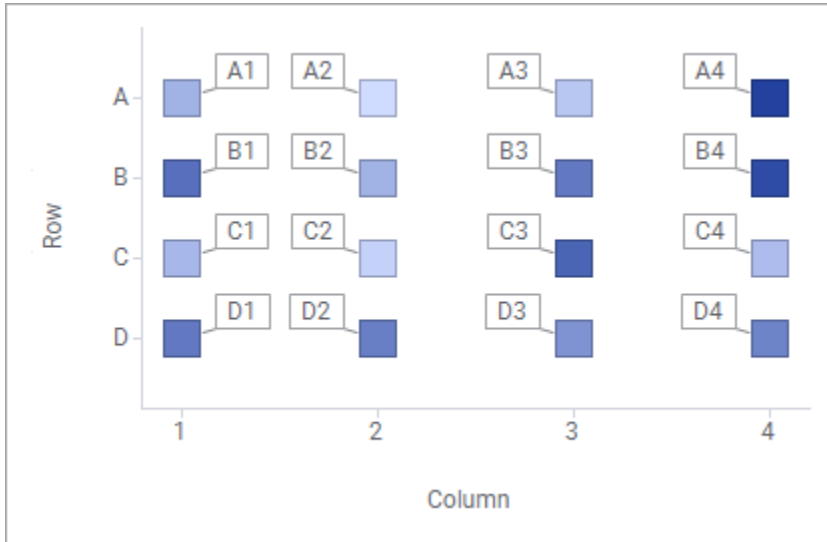


See also [Using tiled markers](#).

## Using tiled markers

A special use of marker shapes in a scatter plot is the tiled markers. Tiled markers are shaped as rectangles, displayed without gaps in a grid, and any added labels are always centered.

For example, if you change the markers in the top scatter plot to tiled markers, you get the scatter plot beneath it.



### Prerequisites

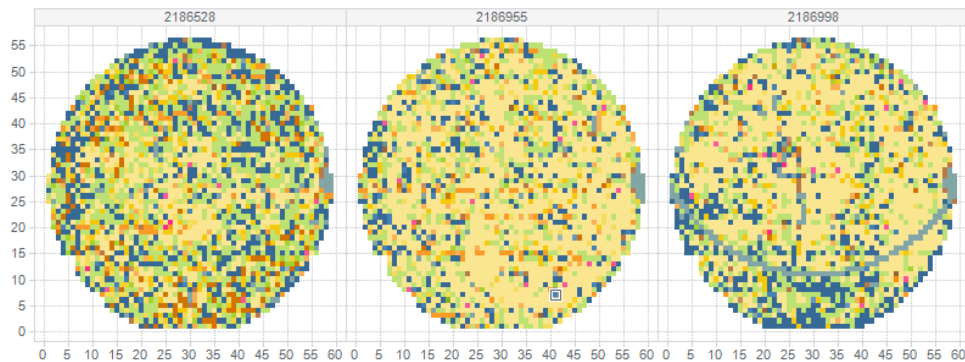
A scatter plot is created.

### Procedure

1. Right-click the scatter plot, and select **Properties**.
2. In the Properties popover, select **Shape**.  
The Shape section is shown.
3. Select **Tiled markers**.

## Example

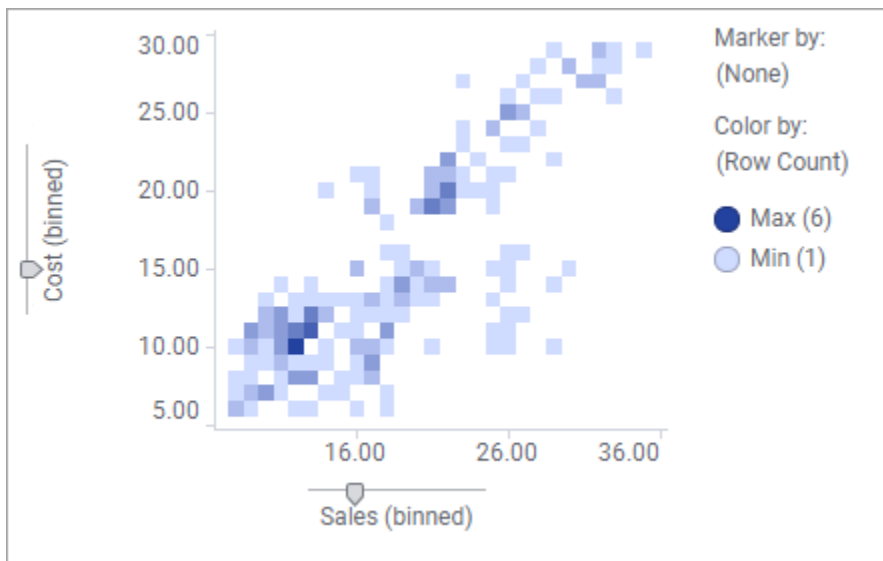
An example of a scatter plot with tiled markers (also known as wafer map) is useful when visualizing and analyzing semi-conductor and plate array tests, because it resembles the actual layout.



For more examples, see [Creating a density plot](#).

## Creating a density plot


The density plot shows how numerical data, binned into intervals, is distributed across the X-axis and Y-axis. To visualize the density, that is, visualize to which extent the markers overlap each other totally, the Color by setting is used.



The markers that represent the binned values are [shaped as tiles](#) in a grid.

A density plot is created from a scatter plot visualization, but a pre-configured density plot can appear as one of the visualizations that is recommended to you, when you select two numerical data columns in the Data in analysis flyout. Then you can add it by simply dragging it to the visualization canvas. If you want to create the density plot from scratch, follow the procedure below.

### Procedure

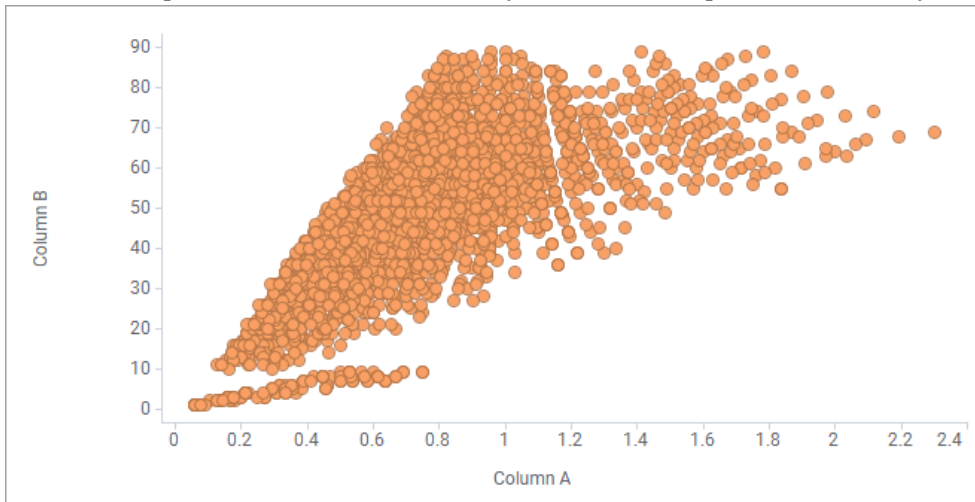
1. On the authoring bar, click **Visualization types**  to open the flyout.



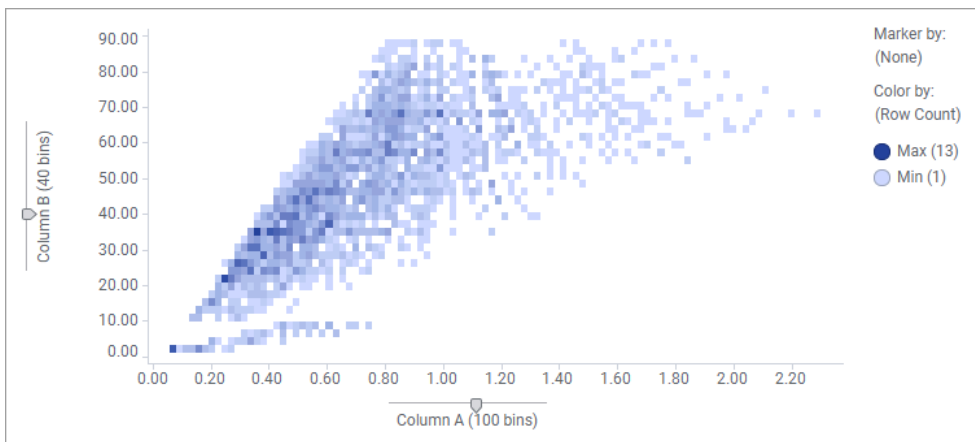
2. Drag the **Scatter plot** visualization type to the wanted position on the analysis page.  
A suggestion of a scatter plot is presented.
3. Select the **numerical column** to display on the X-axis.
4. Select the **numerical column** to display on the Y-axis.
5. **Auto-bin** the X-axis and the Y-axis.
6. Make sure **Marker by** is set to (None).
7. Set **Color by** to (Row count).
8. Right-click the scatter plot, and select **Properties**.  
The Properties popover is shown.
9. Select **Shape**.  
The Shape section is shown.
10. Select **Tiled markers**.
11. Drag the sliders on the axes to adjust the number of bins.

### Example

In the scatter plot below, it seems that many markers overlap each other totally.



If you follow the procedure above, the plot that is created indicates the density by letting colors reflect how many data rows the tiled markers represent.

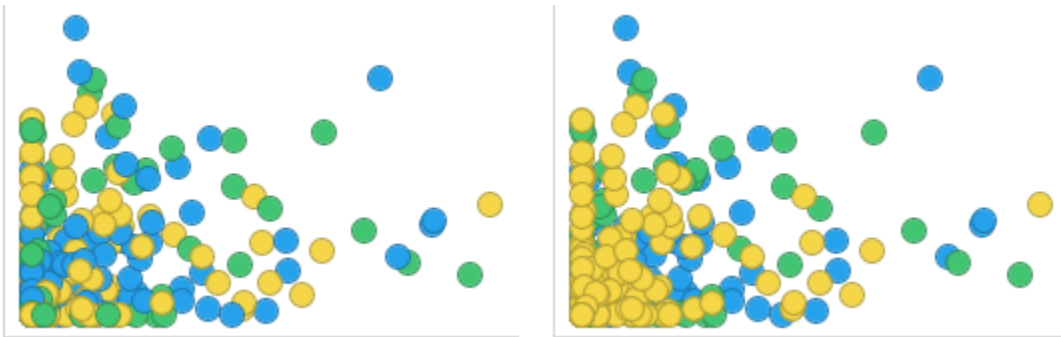


## Adjusting the drawing order of markers

When markers cover each other, partly or totally, it can be useful to specify an order in which the markers are drawn. Markers of certain interest can then be brought forward in the visualization, and other markers backward.

When adjusting the default drawing order, you let the values in a specified data column define the order in which the markers are drawn. For numerical columns, markers representing the lowest values can be drawn first and the highest last or vice versa, and for text-based columns, the drawing order can be alphabetical. However, it is possible to [change the sort order of the values in a column](#).

An example of adjusting the drawing order follows. The visualization to the left shows the default drawing of the markers that are colored by a specified column. The visualization to the right is identical except for the drawing order of markers. By using the color column also for defining a drawing order, the blue markers have been drawn first, then the green markers, and last the yellow markers on top of the others.

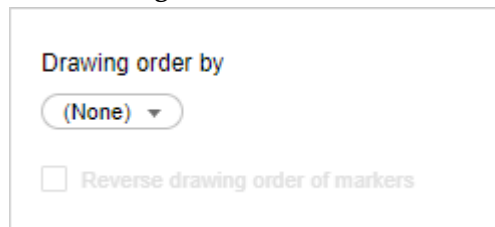


### Prerequisites

A visualization with markers is created. It can be a scatter plot or a map chart with markers.

### Procedure

1. Right-click the visualization with markers, and select **Properties** in the opened menu.  
The Properties pop-over is shown.
2. Click **Drawing order**.  
The **Drawing order** section is shown.



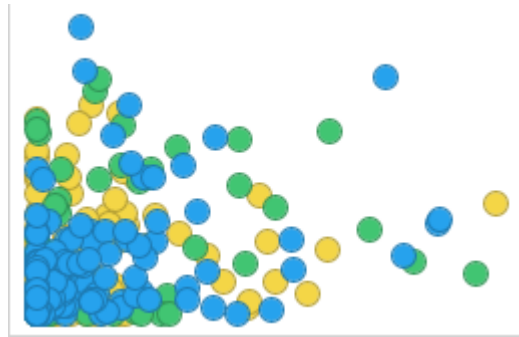
3. In the **Drawing order by** list, select the data column to use for defining in which order markers should be drawn.  
The markers are drawn using the sort order of the column values.

### Result

The markers are drawn using the sort order of the column values.

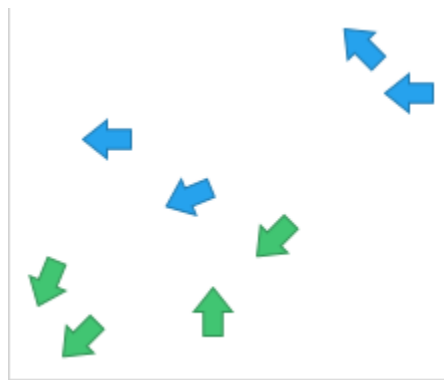
If you want to draw the markers in the opposite order, select the **Reverse drawing order of markers** check box. Front markers will then be sent backwards and vice versa.

Reversing the drawing order in the upper-right visualization will draw the yellow markers first and the blue markers last.



## Showing directions using marker rotation in a scatter plot

Some marker shapes, for example arrows, indicate directions. If you in a scatter plot rotate such markers, you can visualize directions of motions. What you need is a column that contains values that can be interpreted as degrees.



The rotation of the markers can be made clockwise or counter-clockwise. A full rotation is 360 degrees, and a value of 361 is interpreted as 1 degree.

### Procedure

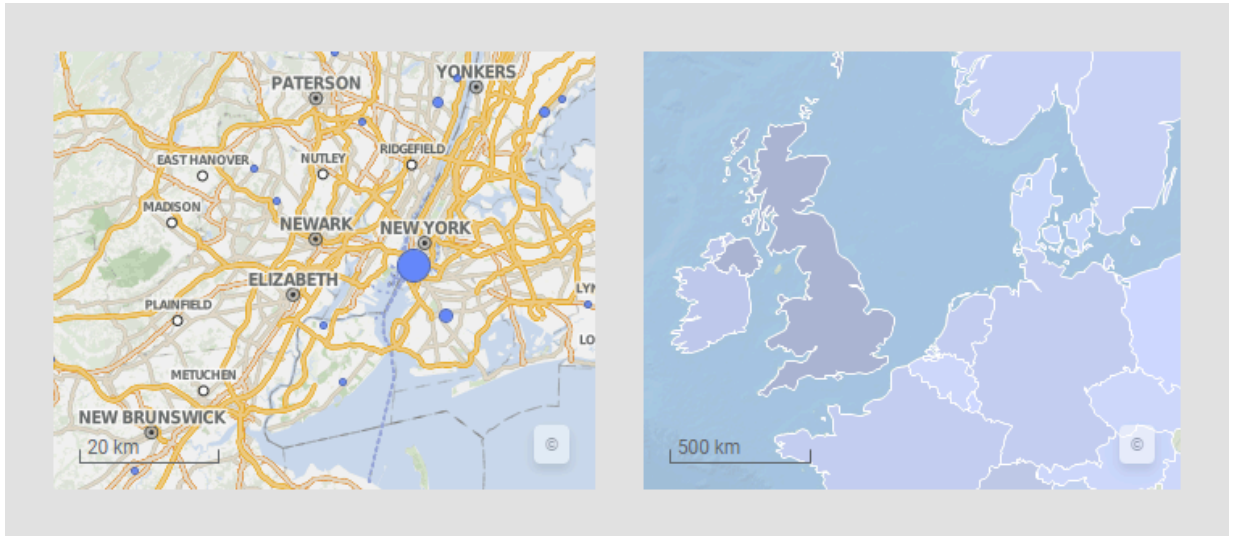
1. Right-click the scatter plot, and select **Properties** in the opened menu.  
The Properties popover is displayed.
2. In the popover, select **Shape**.  
The Shape section is displayed.
3. Select **Fixed shape**, and in the shape palette, click a marker suitable for showing direction.
4. In the popover, select **Rotation**.  
The Rotation section is displayed.
5. Select the column and, if applicable, which type of aggregated value to use for defining how many degrees the markers should be rotated.
6. Select rotation direction; **Clockwise** or **Counter-clockwise**.

### Result

The markers in the scatter plot are rotated to indicate different directions.

## Map chart

A map chart positions your data in a context, often geographical, using different layers. The layers can be either data layers, such as marker layers or feature layers, or reference layers, such as map layers, Tile Map Service layers (TMS), Web Map Service layers (WMS), or image layers. By adding different layers to the map chart, you can configure the map to suit your needs.



The data you use to create a map chart often has geographical context, such as columns with latitude and longitude values, or names of countries and cities.

You can add more than one layer of each layer type and control the properties for each layer separately. Zoom and pan using the [navigation controls](#) in the visualization to explore the data in the map chart.

- Each layer contains its own set of [properties](#) to add fine control for the specific type of layer.
- Transparency, a property in each layer, gives you fine control over the layer display.
- The [marker](#) layers can add indicators for a data column, such as locale or zip codes, major cities, or country names.
- The [feature](#) layers can add shapes defining areas such as regions, countries, or continents.
- The [image](#) layers can be used as both geographical and non-geographical backgrounds in your maps.
- Third-party services ([TMS](#) and [WMS](#)) can be used to add specific field interest or beauty to the map charts.


## Creating a map chart

A map chart positions your data in a context, often geographical, using different layers. The layers can be either data layers, such as marker layers or feature layers, or reference layers, such as map layers, Tile Map Service layers (TMS), Web Map Service layers (WMS), or image layers. By adding different layers to the map chart, you can configure the map to suit your needs.

### Prerequisites

See [Map chart](#) on page 392 for a description of the concept.

### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.

2. Drag the **Map chart** visualization type to the wanted position on the analysis page.  
A suggestion of a map chart is presented.
3. Configure the map chart properties, and then add and configure any number and variety of layers to the map chart.

## Setting map chart properties

The map chart contains properties for title and description, zooming (or controlling navigation), trellis views, and the legend display. Each layer also contains a number of properties specific to the layer. You should review and define the properties for the map chart itself as well as for all layers.

### Prerequisites

Create a map chart visualization.

### Procedure

1. Right-click the map chart visualization, and from the menu, click **Properties**.  
The Properties popover is displayed, with the Map chart popover showing the current layers, in their layer order, top to bottom.
2. Click the label **Map chart** at the top of the layer list to review the generic map chart properties.  
The Properties popover for the map chart is displayed.
3. Set the properties of interest to control the map chart display.
4. Click on the different layers in the layer list to review or edit layer specific properties.

## Adding layers

A map chart is normally built by several different layers. A number of different types of layers exist and each layer type serves a different purpose in the map chart.

Add the relevant layers to your map chart and configure them to suit your needs. You can add a layer either by creating a new layer and configuring it from start, or you can reuse the settings from an existing layer by duplicating it.

## Adding a marker layer

Markers display positional coordinates on a map. The markers are customizable shapes that are placed on the map based on the position coordinates (latitudes and longitudes), or based on the geographic information provided in your data that represent zip codes, cities, counties, regions, or countries.

### Prerequisites

The data must contain geographical or positional information. If the data contains no geographic coordinates (latitude and longitude), markers can be placed automatically in the center of supported administrative areas.

### Procedure

1. Right-click the map chart visualization, and from the menu, click **Properties**.  
The Properties popover is displayed, with the Map chart popover showing the current layers, in their layer order, top to bottom.
2. Click **Add new layer** at the bottom of the Map chart list.  
A list of the layer options is displayed.
3. From the list, click **Marker layer**.

4. From the menu that opens, select the data table to add markers from; either one of your own data tables at the top of the menu, or one of the available geocoding tables.  
The marker layer is added to the Map chart layer list, the Properties list for the layer is displayed, and the layer with its default settings is added to the map chart.
5. Optional: Drag and drop the layer to a new position on the list.
6. Review properties for the map chart to refine the display.



You can reuse the settings of an already existing layer. Select **Duplicate layer** in the **Properties** popover to duplicate either the selected layer in the same map chart, or to add a copy of a layer from another map chart in the analysis. See [Duplicating a layer](#) on page 416 for detailed instructions.

### Positioning markers on the map

Use the **Geographic location** property settings if you want to make changes to the automatic attempt to place geographical elements. Select **Name-based geocoding** to place markers according to the geographical name in the data. The markers for the geographical names are automatically positioned (by means of the geocoding tables) on their geographical coordinates on the map. For more information on geocoding, see [Geographic location and geocoding](#) on page 169. Select **Coordinate-based** to place markers using columns in your data that contain coordinates.

### Marker layer example

This map chart shows Sweden with a nearly-transparent feature layer, showing county boundary lines, and a marker layer with the following properties.

- **Title** is set to Sweden county capitals.
- **Appearance** is set to opaque.
- **Data** is set to Sweden county capitals, with marking.
- **Geographical location** is set to Coordinate-based.
- **Shape** is set to **Shape by column values**, and the column set to cities larger than 200,000. (Three cities, Stockholm, Malmö, and Göteborg meet this condition (TRUE), so their shapes are filled circles, and all other county capitals are filled squares.)
- **Legend** is set to display, with **Marking**, **Color by**, **Shape by**, and **Size by** available.
- **Color** is set to **Color scheme** with two colors, **Color by** set to Sum(Population 2015) so smaller cities and towns appear in blue, and the larger cities appear in red.
- **Size** is set to **Size by** Sum(Population 2015) with marker size slightly enlarged. (Stockholm's marker is significantly larger but the smallest markers are still clearly visible.)
- **Drawing order** is not set because the map has no overlapping markers.
- **Jittering** is not set because the map has no overlapping markers.
- **Line connection** is not set.
- **Labels** is set to display the names of all county capitals.



### ***Showing directions using marker rotation in a map chart***

Some marker shapes, for example arrows, indicate directions. If you, in a map chart marker layer, rotate such markers, you can visualize directions of motions. What you need is a column that contains values that can be interpreted as degrees.

The rotation of the markers can be made clockwise or counter-clockwise. A full rotation is 360 degrees, and a value of 361 is interpreted as 1 degree.

#### **Prerequisites**

A map chart with a marker layer has been created.



## Procedure

1. Right-click the map chart, and select **Properties** in the opened menu.  
The Properties popover is displayed with a side panel showing the current layers in the map chart.
2. If the popover does not display the properties for the marker layer of interest, select the marker layer in the side panel.  
The Properties popover for the selected marker layer is displayed.
3. In the popover, select **Shape**.  
The Shape section is displayed.
4. Select **Fixed shape**, and in the shape palette, click a marker suitable for showing direction.
5. In the popover, select **Rotation**.  
The Rotation section is displayed.
6. [Select the column](#) and, if applicable, [which type of aggregated value](#) to use for defining how many degrees the markers should be rotated.
7. Select rotation direction; **Clockwise** or **Counter-clockwise**.

## Result

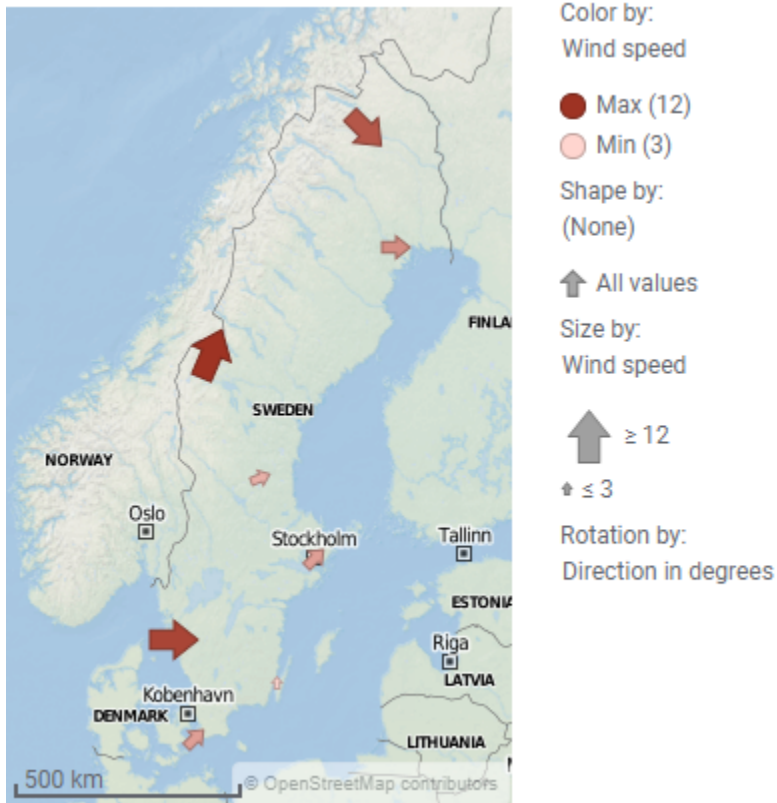
The markers in the marker layer are rotated to indicate different directions.





### Map chart marker layer showing wind direction

The data table below lists wind speed and wind direction, for a number of locations. The wind directions are expressed in degrees.

Location	Longitude	Latitude	Wind speed	Direction in degrees
Kiruna	20.34	67.83	10	135
Boden	21.64	65.82	7	90
Korsvattnet	13.50	63.84	12	23
Edsbyn	15.72	61.36	5	68
Stockholm	18.06	59.63	6	45
Göteborg	11.95	57.75	11	90
Falsterbo	12.82	55.38	6	45
Kalmar	16.39	56.69	3	0

The markers in the marker layer below are used to visualize the direction of the wind at each location (and the marker sizes reflect the wind speed).



If the selected marker shape is , then, a 90 degree clockwise rotation of this marker shape will be displayed as . If you instead select  as the marker shape to use, a 90 degree clockwise rotation would be displayed as .

How many degrees each marker should be rotated is defined by the 'Direction in degrees' data column. The rotation direction in this example is set to clockwise.

## Adding a feature layer

Features display areas or objects on a map. The features are usually administrative areas that are placed on the map based on the geographic information provided in your data and they can represent things like zip codes, cities, counties, regions, or countries. When shapefiles or GeoJSON files are opened in Spotfire they are automatically configured so that they can be used as feature layers in map charts. However, there may be times when some manual work is needed before the data can be used in a feature layer.

### Procedure

1. Right-click the map chart visualization, and from the menu, click **Properties**.  
The Properties popover is displayed, with the Map chart popover showing the current layers, in their layer order, top to bottom.
2. Click **Add new layer** at the bottom of the Map chart list.  
A list of the layer options is displayed.
3. From the list, click **Feature layer** and select the data table to draw the features from; either one of your own data tables at the top of the menu, or one of the available geocoding tables.  
For more information on geocoding, see [Geographic location and geocoding](#) on page 169.  
The feature layer is added to the Map chart layer list, the Properties list for the layer is displayed, and the layer with its default settings is added to the map chart.
4. Optional: Drag and drop the layer to a new position on the list.
5. Review properties for the map chart to refine the display.



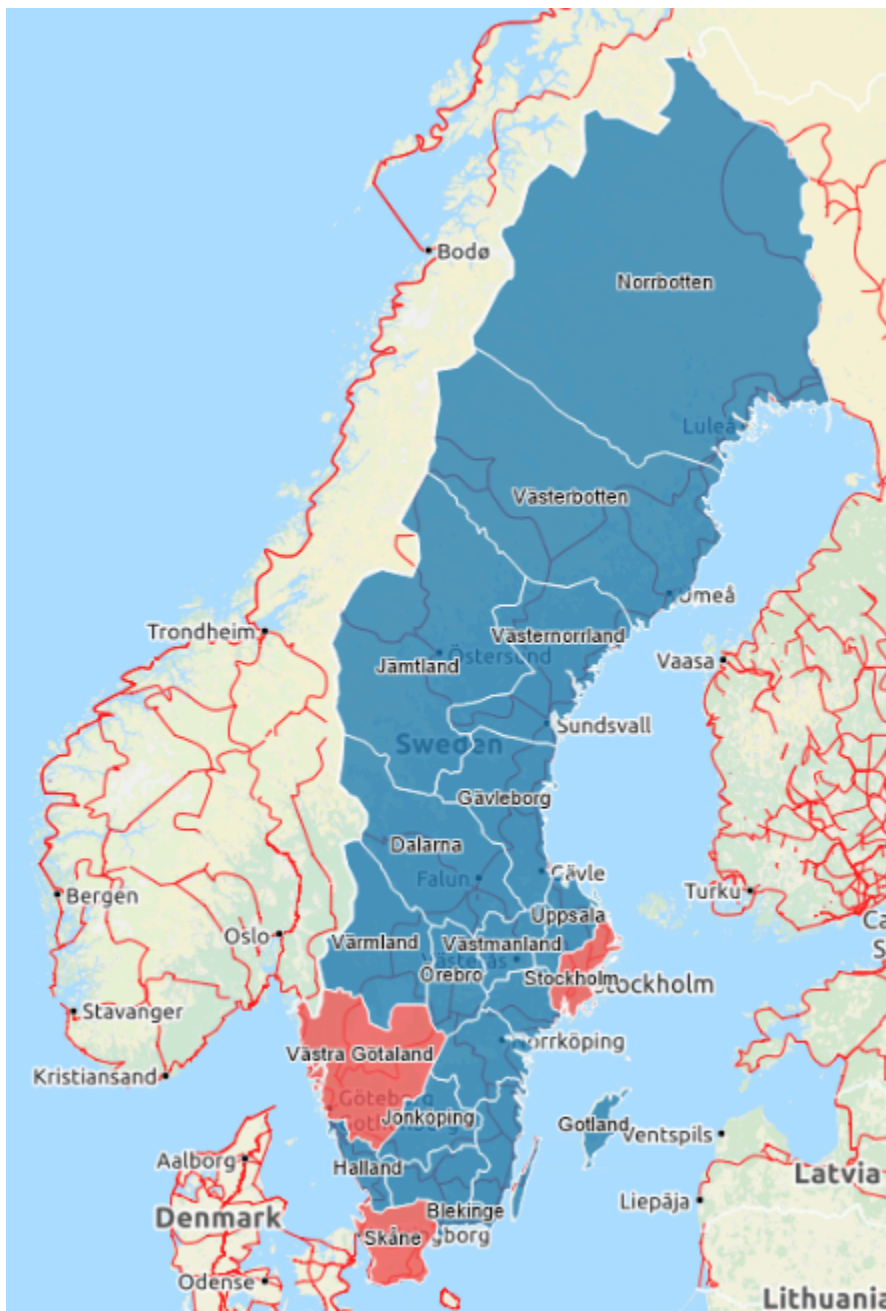
You can reuse the settings of an already existing layer. Select **Duplicate layer** in the **Properties** popover to duplicate either the selected layer in the same map chart, or to add a copy of a layer from another map chart in the analysis. See [Duplicating a layer](#) on page 416 for detailed instructions.

### Feature layer example

This map chart visualization shows Sweden, using two data tables: Sweden Counties and Sweden county capitals. The data includes added census information from 2015, and it has a calculated column indicating cities with populations over 200,000. The map chart visualization includes a feature layer with the following properties.

- **Title** is set to `Sweden Counties (Län)`
- **Appearance** is set to semi-transparent (displaying the OpenStreetMap cycling map TMS layer beneath it).
- **Data** is set to the data table `Sweden county capitals - Swedish Counties (Län)`.
- **Geographic location** is set to **Feature by** `County (län)`.
- The **Legend** check box is cleared.
- **Color** is set to the following values.
  - **Color by:** `Larger than 200000` (an added calculated column).
  - **Color scheme:** 2 colors.
  - **Color items:** `True` (over 200,000) and `False` (under 200,000), with color selections customized to blue and red.
- **Labels** is set to `First(County (län))`.

The result is that the counties with cities over 200,000 (Stockholm, Västra Götaland, and Skåne) are the first color, while all other counties are the second color.



You can add your own features. For information about creating your own features, see the following articles. (These articles are in the [Community](#) site. It is free to use but requires log in credentials.)

- [Expanding Spotfire geocoding coverage](#)
- [Where to find geographic data sources for Spotfire?](#)

## Requirements and settings for feature data in Spotfire

This bullet list is a summary of what is needed to view interactive shapes in Spotfire. See also [Configuring WKB data for use with maps](#).

- The binary geometry data must be extracted to one Geometry column and six coordinate columns. The columns also need to be assigned the proper value on the **MapChart.ColumnTypeId** column property: Geometry, XMin, XMax, YMin, YMax, XCenter and YCenter.
- The **ContentType** column property must be set to **application/x-wkb** for the Geometry column.
- The data type of the geometry column should be "Binary" or "BLOB".
- The data types for the coordinate columns should be "Real".
- The renderer for the geometry column should be set to **Geometry** in the Visualization Properties of the selected visualization (available in the installed client only). For example, in the Columns tab of a Table visualization.

You can also show geometries as shapes in some labels and tooltips, and might also need to specify the Geometry renderer there.

## Adding an image layer

With an image layer, you can configure your map chart to use images in combination with both geographical and non-geographical data. For example, you can add a background image and position markers on top of the image.

### Procedure

1. Right-click the map chart visualization, and from the menu, click **Properties**.  
The Properties popover is displayed, with the Map chart popover showing the current layers, in their layer order, top to bottom.
2. Click **Add new layer** at the bottom of the Map chart list.  
A list of the layer options is displayed.
3. From the list, click **Image layer**.  
The image layer is added to the Map chart layer list, and the Properties for the layer is displayed.
4. In the **Image** section, click **Browse**.
5. Navigate to the image you want to use, and add it to your map chart.  
The image is loaded.



Sometimes the image you have selected does not contain geographical metadata. In those cases a dialog might appear, informing you that the coordinate reference system must be 'None' in all parts of the map chart visualization if you are going to use the image as a background. Click **Yes** if you want to use the image as a background. The coordinate reference system will then be set to 'None' right away. Click **No** if the image should not be used as a background image. See [Moving and resizing images](#) for examples where non-geographical images are not used as backgrounds in maps.

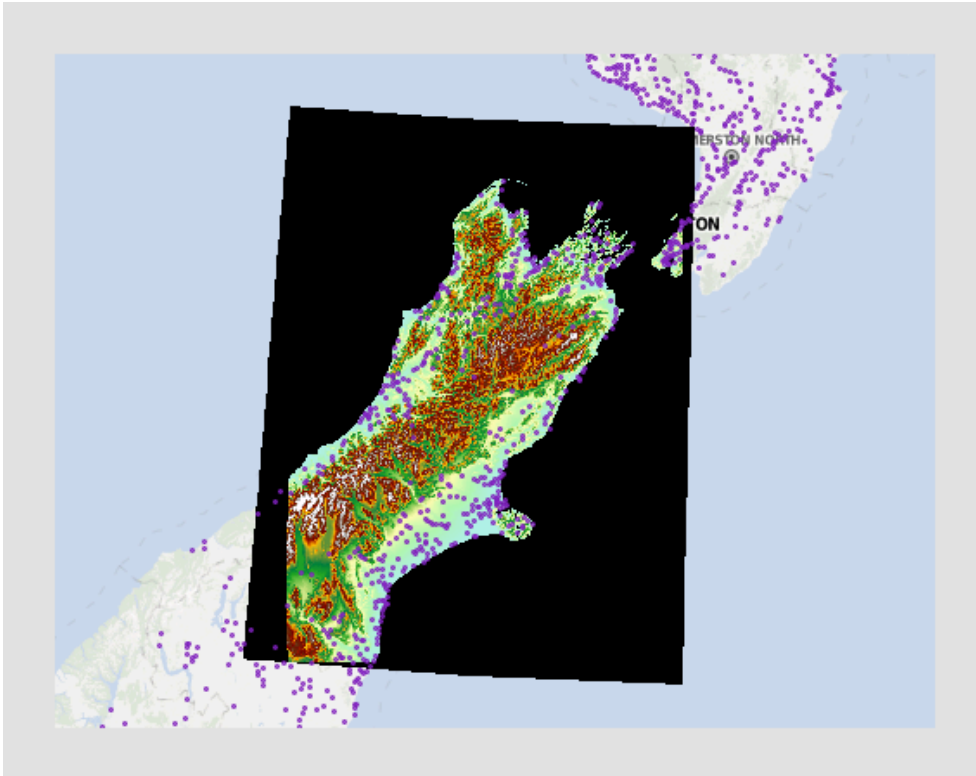
6. Optionally, drag and drop the layer to a new position on the list.



You can reuse the settings of an already existing layer. Select **Duplicate layer** in the **Properties** popover to duplicate either the selected layer in the same map chart, or to add a copy of a layer from another map chart in the analysis. See [Duplicating a layer](#) on page 416 for detailed instructions.

### Image layer examples

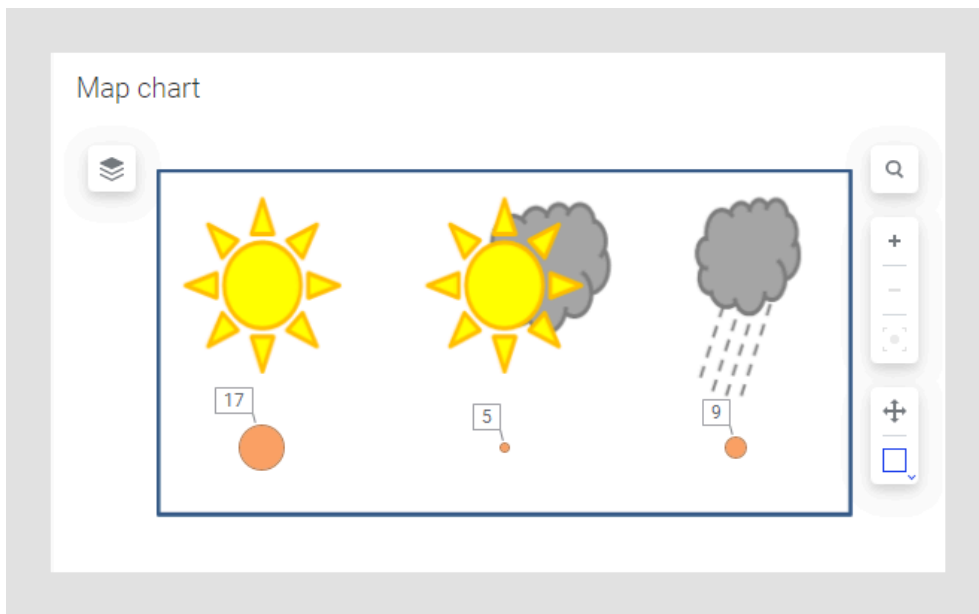
Using a background image works similarly to the map with markers, but with the difference that you do not need to have map data in a data table in order to set it up. If you use a GeoTIFF image, where geographical information is included as metadata in the image file, then the image will automatically be positioned correctly. The example below shows a GeoTIFF image that is projected onto a map layer, and the markers are placed on top of the image layer.



Note that for the GeoTIFF image to be placed correctly on the map, the image must include projection metadata, and that some projections are not supported.

With other types of image files, the data table must contain X and Y coordinates for the markers to be placed correctly in geographical positions.

A map chart can also be used to show non-geographical data. The example below shows a map chart where the background is an image illustrating the three weather types sunny, partly cloudy, and rainy. The markers placed on top of the image show how many days in the month of August that were sunny, partly cloudy, and rainy respectively.



See [Positioning markers on an image background](#) to learn more about this example.

You can choose how you want the image to be displayed when it is scaled up compared to its original size and resolution. Select the check box **Preserve sharp edges when upscaling image** in the **Appearance** section on the **Properties** popover if you want to keep sharp (pixelated) edges, and avoid smoothing, when scaling up the image.

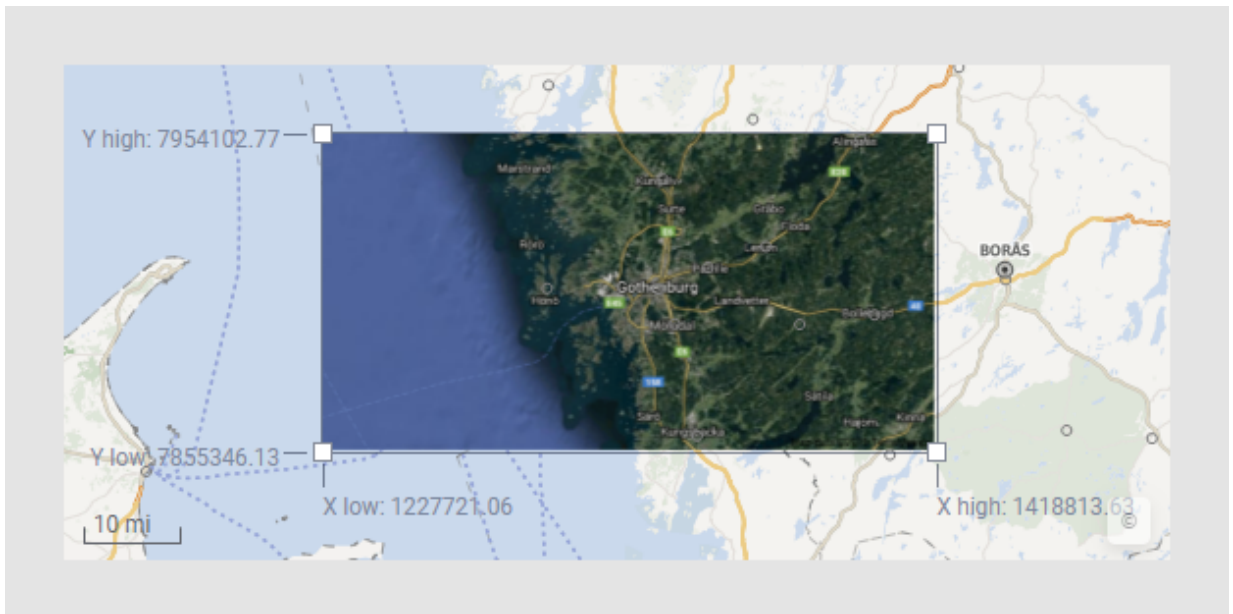
The example below shows the difference between the two options in an upscaled image. In the image to the left, the check box is cleared. In the image to the right, the check box is selected.



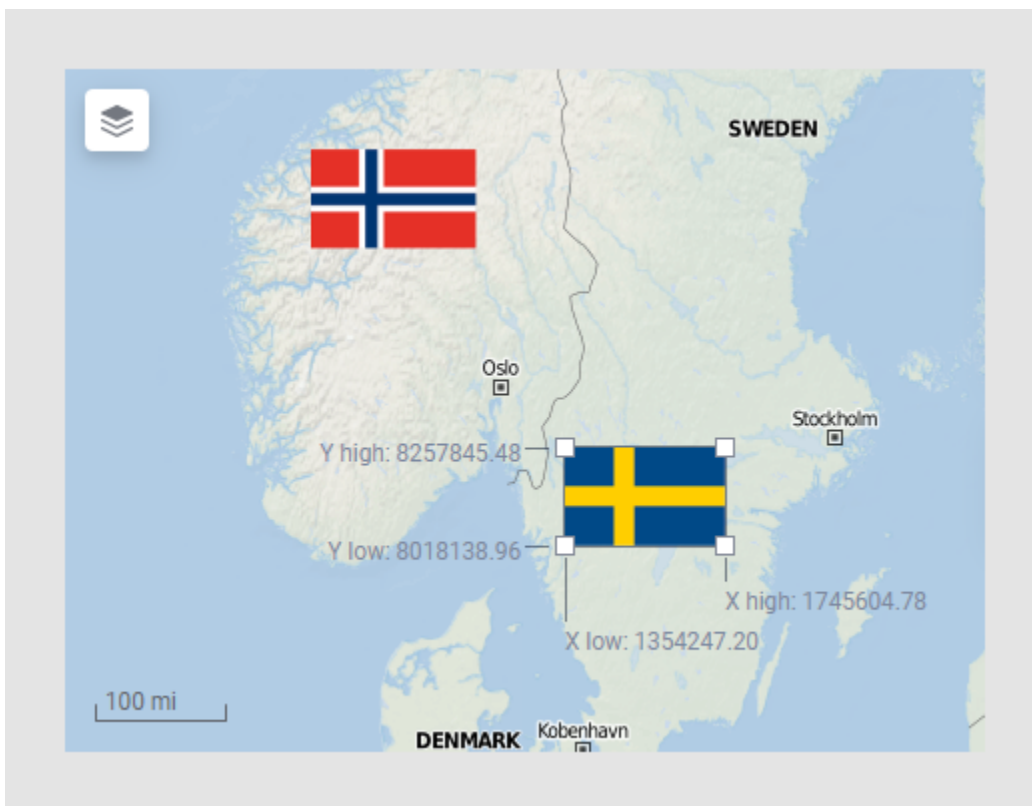


## Moving and resizing images

Images in image layers can be moved and resized on top of the map chart directly. The possibility to move images can be useful when you, as illustrated below, want to position a satellite photo on top of an ordinary map layer, or when you have a number of images to arrange on top of a map.



The image and map should use identical coordinate reference systems to match properly.



## Prerequisites

The map chart contains a map layer and an image layer containing an image that is not used as background. One of the following coordinate reference systems is used: EPSG:3857, EPSG:4326, EPSG:4269, or EPSG:4267.

## Procedure

1. Right-click the image, and select **Edit image position** in the menu.  
The image gets a frame with handles in the corners. Coordinates for the corners are displayed.
2. Click and drag the image to the wanted position.
3. If you want to resize the image, drag one of the handles. This will keep the proportions of the image. To change the proportions, drag any of the borders.
4. Right-click the image once again, and deselect **Edit image position**.

## Positioning markers on an image background

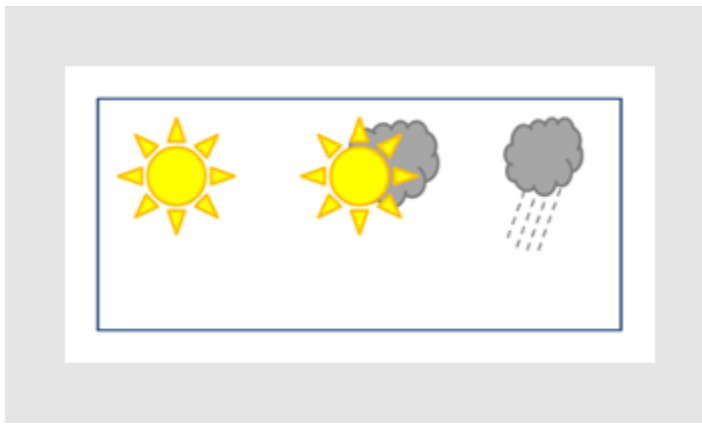
The background of the markers in a map chart does not have to be geographical; you can position markers on any spatial background. You can, as the background to the markers, simply use an image layer instead of a map layer. To exemplify, your image in the image layer can represent a human body, where you place markers showing medical data, or a drawing of a hotel, where the room occupancy or other data is displayed. To position markers on top of an image layer requires that your data contains appropriate X and Y coordinates.

## Example

The data table below contains weather data at a certain location during August. The weather types are Sun, Partly clouded, and Showers.

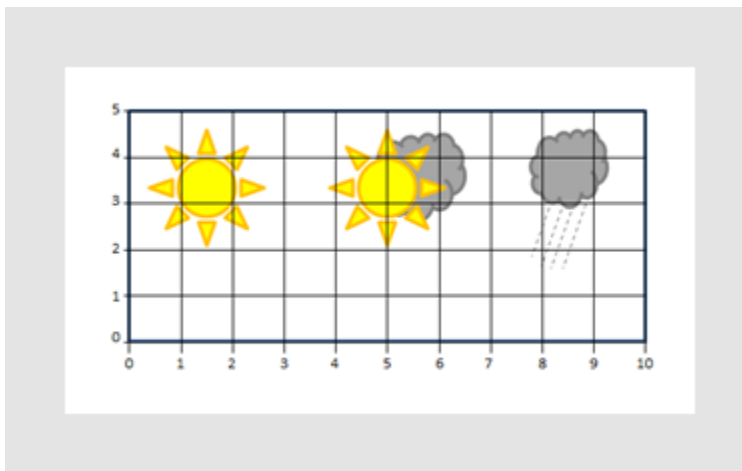
Date	Weather
8/1/2019	Showers
8/2/2019	Showers
8/3/2019	Sun
8/4/2019	Sun
8/5/2019	Showers
8/6/2019	Sun
8/7/2019	Sun
8/8/2019	Sun
8/9/2019	Sun
8/10/2019	Partly clouded
8/11/2019	Partly clouded

Assume you want to use the image below to illustrate the weather types, and, below the illustrations, place markers that indicate the total number of sunny, partly clouded, and rainy days, respectively.



The following explains how you place the markers where you want them on the image. The key is to let the image and markers use a common coordinate reference system.

To do so, imagine a coordinate system on top of the image as shown below. You can specify any X and Y ranges provided that the same scale is used on both axes. In this imagined coordinate system, the X values range from 0 to 10, and the Y values from 0 to 5.



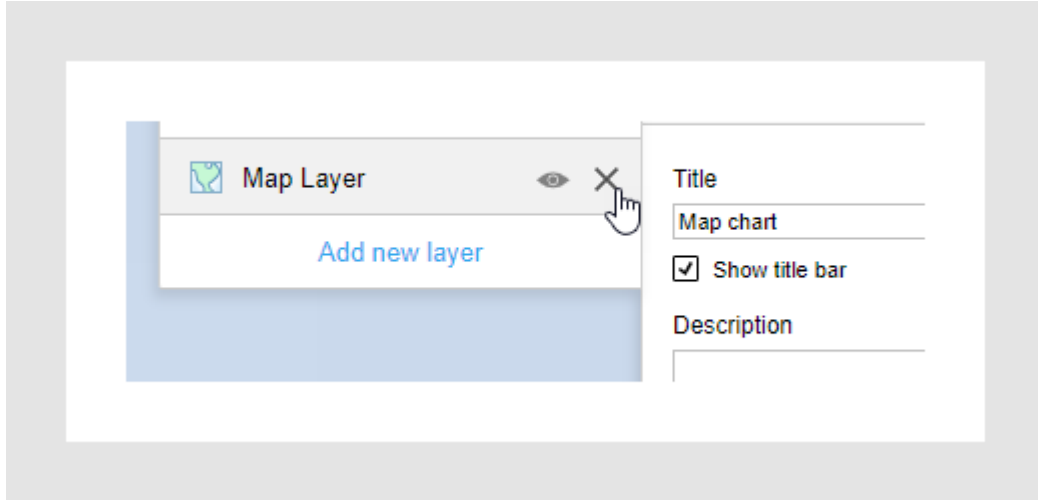
Then use this imagined coordinate system to assign coordinates to your marker positions, and add them as X- and Y columns to the data table. The 'Sun' marker is placed at (1.5,1), the 'Partly clouded' marker at (5,1) and the 'Showers' marker at (8.3,1).

Date	Weather	X	Y
8/1/2019	Showers	8.30	1
8/2/2019	Showers	8.30	1
8/3/2019	Sun	1.50	1
8/4/2019	Sun	1.50	1
8/5/2019	Showers	8.30	1
8/6/2019	Sun	1.50	1
8/7/2019	Sun	1.50	1
8/8/2019	Sun	1.50	1
8/9/2019	Sun	1.50	1
8/10/2019	Partly clouded	5.00	1
8/11/2019	Partly clouded	5.00	1

Now all is set for creating the map chart, where markers are positioned on top of an image.

## Procedure

1. Load the data table, and create a map chart.
2. Right-click the map chart, and select Properties.  
The current layers and the Properties popover are displayed.
3. In the list of layers, remove the Map Layer by placing the mouse pointer on it, and click X.



4. Click **Add new layer**, and select **Image layer**.  
The Properties popover displays the Image Layer settings.
5. In the Image section, via **Browse**, add the image you want to use as background.
6. In the Add image as background dialog, click **Yes** to set the coordinate reference systems to None.
7. Open the Positioning section in the Properties - Image Layer.

8. Beneath **Position settings**, enter the max and min of the X- and Y values that are used to specify the imagined coordinate reference system, that is, the coordinate reference system in which the markers' coordinates are set.


**Positioning**

Coordinate reference system

None ▼

Other...

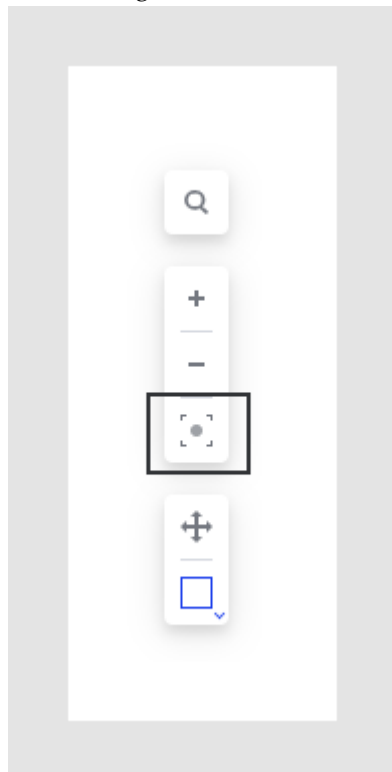
Position settings



Y Low	Y High
0	5
X Low	X High
0	10

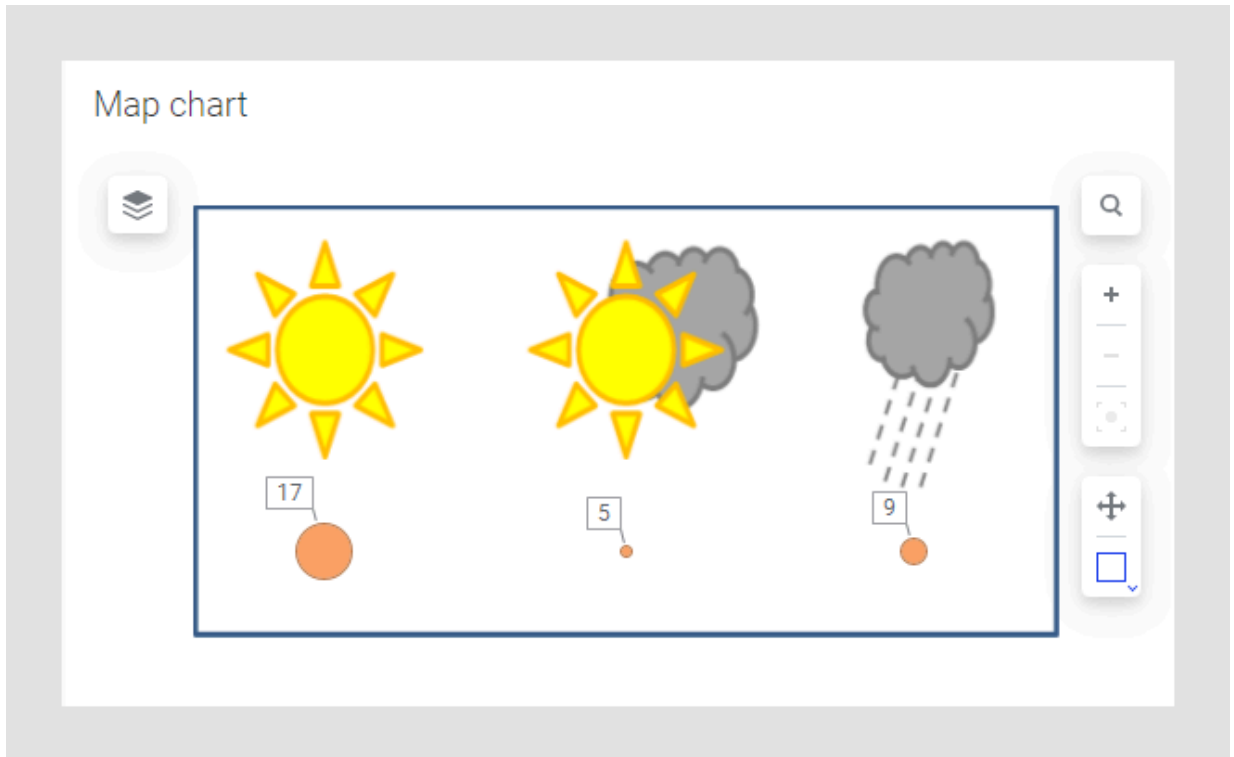
Reset

9. In the navigation control of the map chart, click **Zoom to filtered items**, if needed.



## Result

The result is shown below. In addition to the steps above, the labels and sizes of the markers are set to Count(Weather).



## Adding a map layer

A map layer gives a geographical context to your data by rendering a worldwide map background under your data layers. You can add multiple map layers in one map chart.

### Procedure

1. Right-click the map chart visualization, and from the menu, click **Properties**.  
The Properties popover is displayed, with the Map chart popover showing the current layers, in their layer order, top to bottom.
2. Click **Add new layer** at the bottom of the Map chart list.  
A list of the layer options is displayed.
3. From the list, click **Map layer**.  
The map layer is added to the Map chart layer list, the Properties list for the layer is displayed, and the layer with its default settings is added to the map chart
4. Optional: Optionally, drag and drop the layer to a new position on the list.
5. Review and set the map chart properties of interest to refine the display.

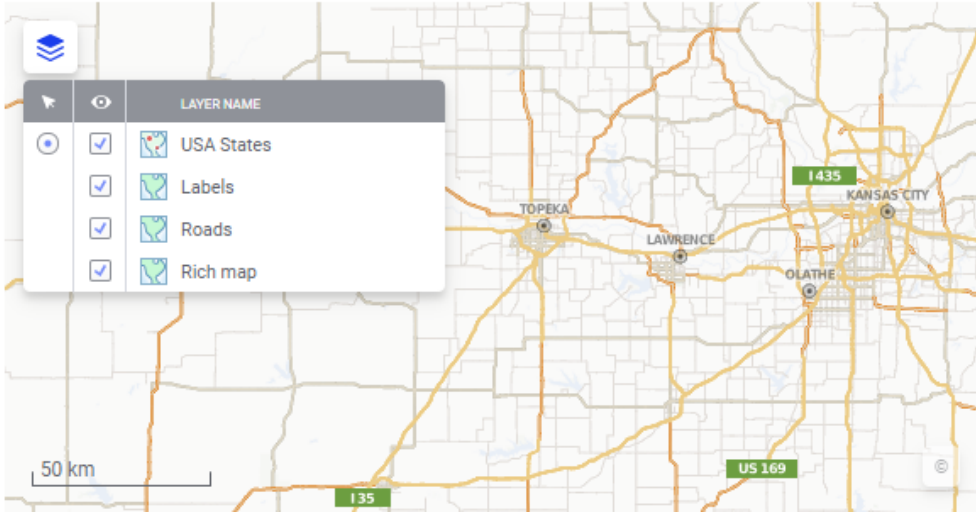


You can reuse the settings of an already existing layer. Select **Duplicate layer** in the **Properties** popover to duplicate either the selected layer in the same map chart, or to add a copy of a layer from another map chart in the analysis. See [Duplicating a layer](#) on page 416 for detailed instructions.

## Map layer example

This example shows a map chart with multiple map layers. The **transparency** value in the Appearance settings has been modified on the Labels and Rich map layers, to enhance the roads on the map.

### US roads



You can use several map types together, along with transparency settings, for effective layering. For example, you might add four map layers on top of each other, where the bottom layer is a rich map (which does not include any borders, roads, or labels), and the other three are borders, roads, and labels respectively. You can then fine-tune the appearance of the map chart by changing transparency, map style, or simply hiding the layers separately.

## Adding a TMS layer

The Tile Map Service (TMS) is a widely supported map protocol. Use it to display a map background from a third party provider or to change the default map layer.

### Prerequisites

To add a TMS layer, you must have a URL to a valid TMS map.


You can find many TMS resources on the internet. The article [GeoAnalytics Resources](#) on the [Community](#) site lists many TMS resources, both free and paid. The lists include the complete URL, copyright text, and if available, copyright URL to specify in the TMS layer. (The Community site is free to use; however, you must register an account to gain access to the information.)

### Procedure

1. Right-click the map chart visualization, and from the menu, click **Properties**.  
The Properties popover is displayed, with the Map chart popover showing the current layers, in their layer order, top to bottom.
2. Click **Add new layer** at the bottom of the Map chart list.  
A list of the layer options is displayed.

- From the list, click **TMS layer**.

The TMS layer is added to the Map chart layer list, the Properties list for the layer is displayed, and the layer with its default settings is added to the map chart.

An error icon (  ) is displayed in the title bar. (A TMS layer is not valid until the URL to a valid service is supplied.)

- Required: Provide the URL to a valid service.  
The error icon displayed in the title bar is no longer displayed.



The URL must be in the format `http://base/url/{z}/{x}/{y}`. {z} specifies zoom level, {x} specifies the column number, and {y} specifies the row number. You do not need to provide values for the three variables {z}/{x}/{y} for the map to work.

- Optional: Optionally, drag and drop the layer to a new position on the list.

- Review and set map chart properties to refine the display.

Note that it is mandatory to include copyright information for the vast majority of third-party map providers. Open the **Copyright** section of the **Properties** popover to configure the copyright info.

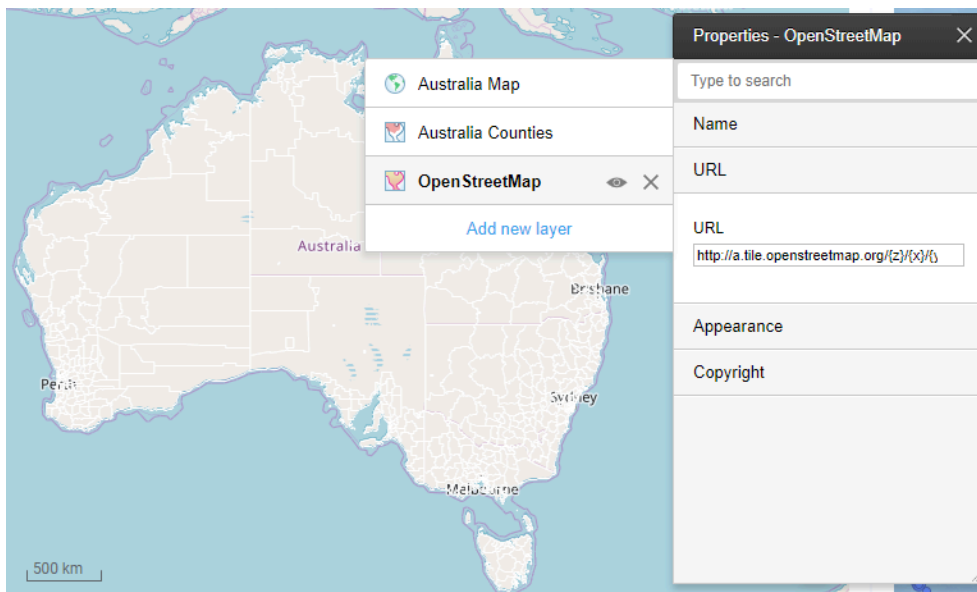


You can reuse the settings of an already existing layer. Select **Duplicate layer** in the **Properties** popover to duplicate either the selected layer in the same map chart, or to add a copy of a layer from another map chart in the analysis. See [Duplicating a layer](#) on page 416 for detailed instructions.

### TMS layer example

In this example, the map chart has a feature layer displaying the outlines of Australia counties on top of a TMS layer.

- The feature layer for Australia Counties is available in the [geocoding](#) data tables provided in the library. The transparency is set to a high degree to show outline and shading. The feature layer is the top layer of the map chart.
- The standard map layer is removed. The TMS layer replaces the standard map.
- The TMS layer is drawn from a map from OpenStreetMap, so the **Title** property is set to OpenStreetMap.
- The **URL** is set to `http://a.tile.openstreetmap.org/{z}/{x}/{y}.png`.





## Adding a WMS layer

The Web Map Service (WMS) is a standard protocol for serving map images that a map server generates using data from a Geographic Information System (GIS) database. You can add a WMS layer to a map chart visualization to provide specialized context.

### Prerequisites


To add a WMS layer, you must have a URL to a valid WMS 1.1.1 (or earlier) service that uses WGS84 coordinate reference system (EPSG:3857 or EPSG: 4326). You can use a WMS server that requires login using basic authentication or that requires no login credentials.



Many WMS resources are available on the internet. The article [WMS sources for Spotfire](#) on the [Community](#) site lists many links to sites that contain WMS resources, both free and paid. (The Community site is free to use; however, users must register an account to gain access to the information.)

### Procedure

1. Right-click the map chart visualization, and from the menu, click **Properties**.  
The Properties popover is displayed, with the Map chart popover showing the current layers, in their layer order, top to bottom.
2. Click **Add new layer** at the bottom of the Map chart list.  
A list of the layer options is displayed.
3. From the list, select **WMS layer**.  
The WMS layer is added to the top of the Map chart list, the Properties list for the layer is displayed, and the layer with its default settings is added as the top layer of the map chart.

An error icon (  ) is displayed in the title area. (A WMS layer is not valid until a URL to a valid service is supplied. See [Configuring a WMS layer and sublayer](#) for more information.)

4. Required: Provide the URL to a valid service.  
The URL must link to a valid XML structure and a server that supports WMS 1.1.1 or earlier service, and which uses WGS84 coordinate reference system (EPSG:3857 or EPSG: 4326).
  - If you supply a link to a server that supports only WMS 1.3.0 or later, an error is displayed in the title area.
  - Most WMS maps are designed for specific regions or countries. If you provide a URL that contains a map structure for another country or region, Spotfire displays an error.

If the specified WMS service requires login, select the check box **Use login credentials**, then specify username and password, and click **Log in**.

When a valid URL to a supported server version is supplied, the error icon in the title bar is no longer displayed.

5. Optional: Drag and drop the layer to a new position on the list.
6. Review properties for the map chart to refine the display.  
See [Configuring a WMS layer and sublayer](#) for detailed guidance and an example.



You can reuse the settings of an already existing layer. Select **Duplicate layer** in the **Properties** popover to duplicate either the selected layer in the same map chart, or to add a copy of a layer from another map chart in the analysis. See [Duplicating a layer](#) on page 416 for detailed instructions.

## Configuring a WMS layer and sublayer

After you have added a valid Web Map Service (WMS) layer, use its properties to fine-tune its display in the map chart visualization.

### Prerequisites

You must have a valid WMS URL. You can use a WMS server that requires login using basic authentication or that requires no login credentials.



Many WMS resources are available on the internet. The article [WMS sources for Spotfire](#) on the [Community](#) site lists many links to sites that contain WMS resources, both free and paid. (The Community site is free to use; however, users must register an account to gain access to the information.)

### Procedure

1. Right-click the map chart visualization, and from the menu, click **Properties**.  
The Properties popover is displayed, with the Map chart popover showing the current layers, in their layer order, top to bottom.
2. From the layers list, select the WMS layer.  
The WMS layer properties are listed in the Properties dialog.
3. Click **Map**.  
The Map property expands to show the **Available sublayers** box.
4. From the list, select the desired sublayers, clicking **Add** for each sublayer to add to the map.  
The layers are added to the map and listed in order in the **Selected sublayers** list.
5. In the **Selected sublayers** list, use the **Move forward** and **Move backward** buttons to change the layering.  
The WMS layer redraws, displaying the sublayers in the order they are listed, top to bottom. If the Legend is displayed, the sublayers are added to the Legend.
6. Optional: Click **Remove** to remove any sublayers that are no longer needed in the map.
7. Set additional display properties, for example transparency and rendering.

## Example of configured WMS layer with sublayers

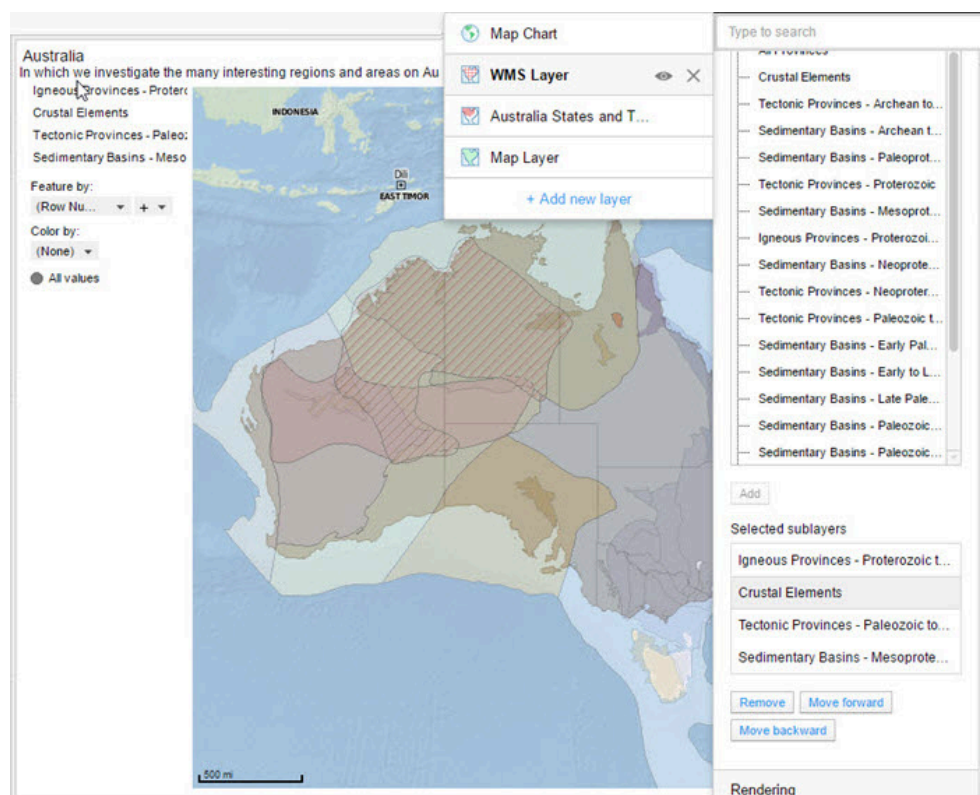
The Australian Geological Provinces WMS layer has many sublayers. In this image the following properties are set.

- The map chart **Title** and **Description** contain information about the visualization.
- The WMS layer is displaying the service from `http://services.ga.gov.au/gis/services/Geological_Provinces_2013/MapServer/WMSServer?request=GetCapabilities`.



Browsing to the link displays the valid XML for the WMS.

- **Appearance** has been set to semi-transparent to display the country and territories under the WMS layer.
- **Map** displays the four selected sublayers (listed in the legend). The selected sublayers top-to-bottom listing corresponds to the forward-to-backward placement on the map chart visualization. Note that the sublayers are also displayed in the legend.
- The legend is displayed on the left side of the map chart visualization.
- The underlying feature layer displays state borders.
- The underlying map layer displays the coastline and ocean.



Some sublayers can totally obscure other layers or sublayers. To be able to show multiple sublayers, consider adding multiple versions of the same WMS layer and changing the **Appearance** property to a higher transparency.

## Rendering tips

On the **Rendering** properties page, you can change properties affecting how fast the map is rendered.

If your WMS server supports tile-base rendering, it will make the map load faster. However, tile-based rendering generates more calls to the WMS server, so if you have a limit on the number of requests, you can run the risk of reaching the limit and not being able to retrieve images from the server. To make the map load faster when the analysis is reopened, make sure to use the setting Cache images from WMS server. This will reduce the number of requests to the WMS server.

## Duplicating a layer

One way to add a layer to a map chart is to duplicate an existing layer. You can duplicate layers from any map chart in the analysis.

### Procedure

1. Right-click the map chart visualization, and from the menu, click **Properties**.  
The Properties popover is displayed, with the Map chart popover showing the current layers, in their layer order, top to bottom.
2. Click **Duplicate layer**. To add a copy of a layer from the same map chart, select the layer you want to duplicate before you click **Duplicate layer**.
3. To duplicate the layer that is currently selected in the Map chart popover, click **Selected layer**. To add a copy of a layer from another map chart, select **Layer from another map chart**, then locate and select the layer you want to duplicate.  
The duplicated layer is added to the map chart.
4. Optionally, drag and drop the layer to a new position on the list.

## Working with layers


From the properties popover, you can refine a map chart by reordering layers, hide layers, or permanently remove layers.

The layers displayed in the map chart, top to bottom, correspond to the order of the layer list in the Map chart properties popover, first to last. In other words, placing a layer first in the layer list will place the layer on top of all other layers in the map chart.

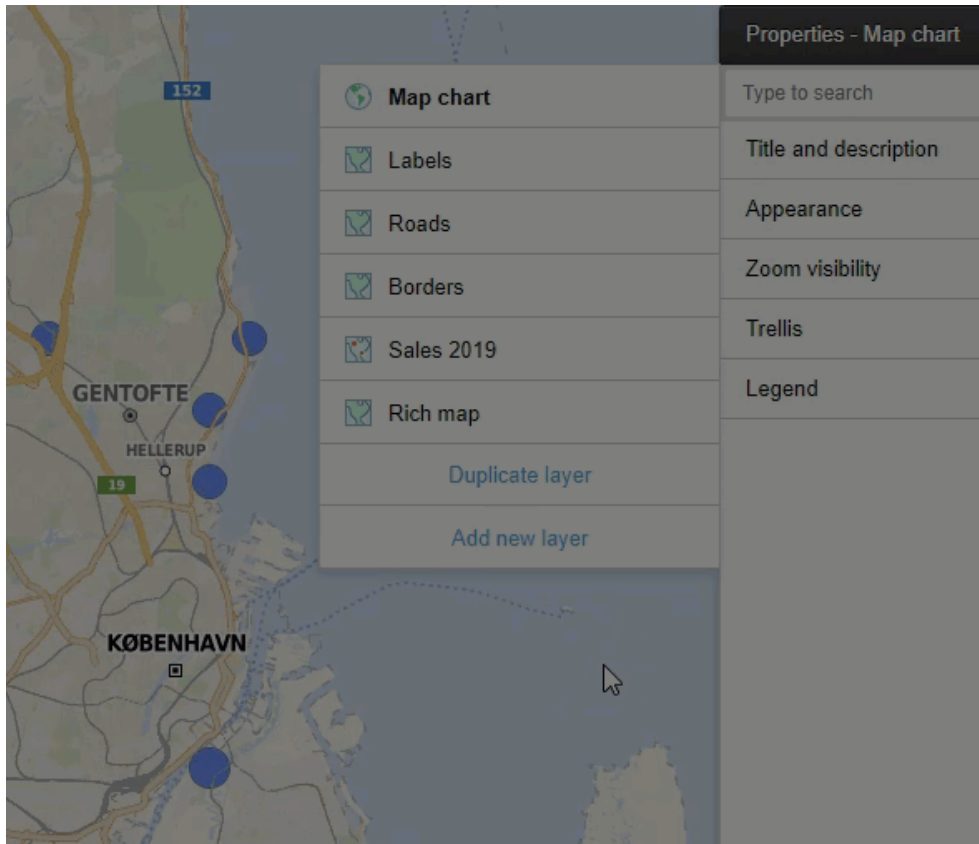
You can hide layers temporarily in a map chart. This can be useful when you are configuring the map chart, because you can see what effect the settings of a specific layer has on the entire map chart. You can also use the Layers control directly in the visualization to hide a layer.

If you no longer need a certain layer, you can remove it permanently.

### Procedure

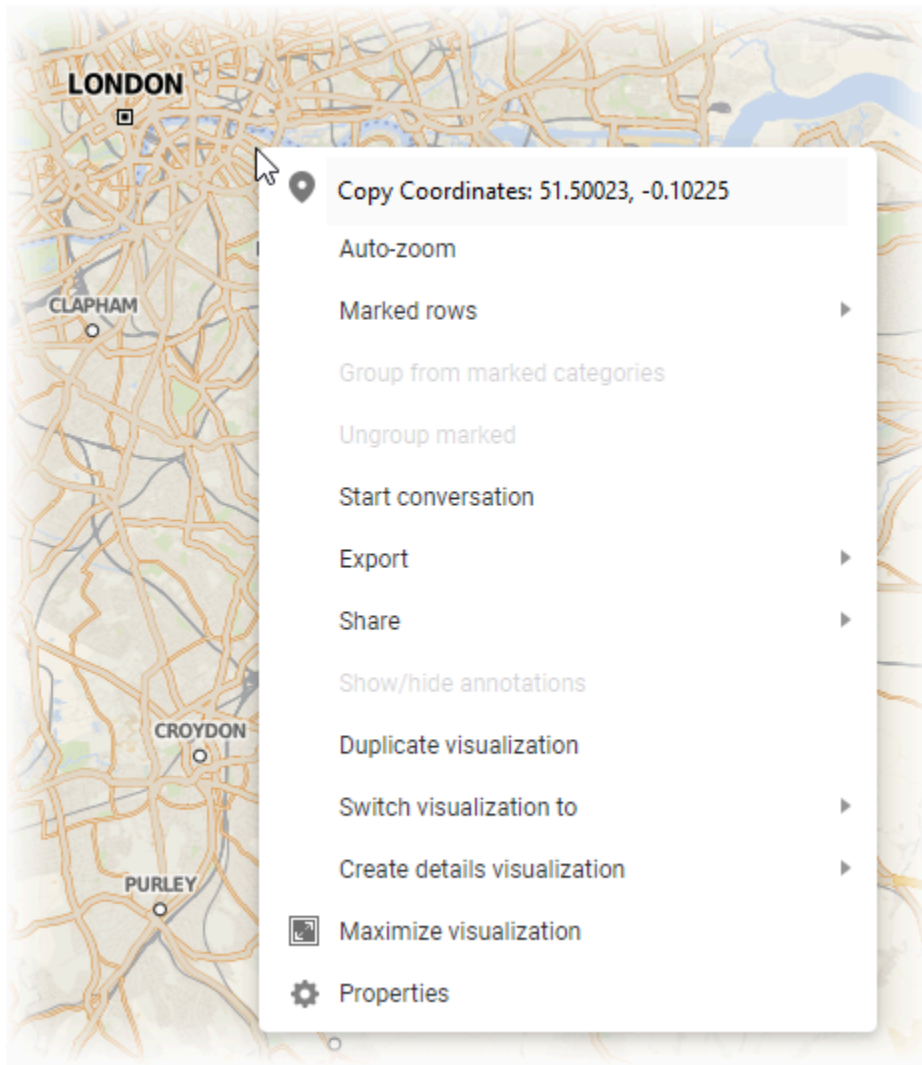
1. Right-click the map chart, and from the menu, select **Properties**.  
The Properties popover is displayed, showing the current layers, in their layer order, top to bottom.
2. Place the mouse pointer on the layer of interest.  
The layer will be highlighted, and icons will become visible.
3. Click the eye icon,  , to hide the layer, click the X to remove the layer permanently, or click and drag the layer up or down to another position.  
The map is updated as you make the changes.

This example first shows how the marker layer is moved first in the layers list so that the markers are placed on top of the map layers Labels, Roads, and Borders. Then the map layer named Roads is temporarily hidden and then shown again.



## Viewing the coordinates for a position

You can view the coordinates for a position in the map chart.



### Procedure

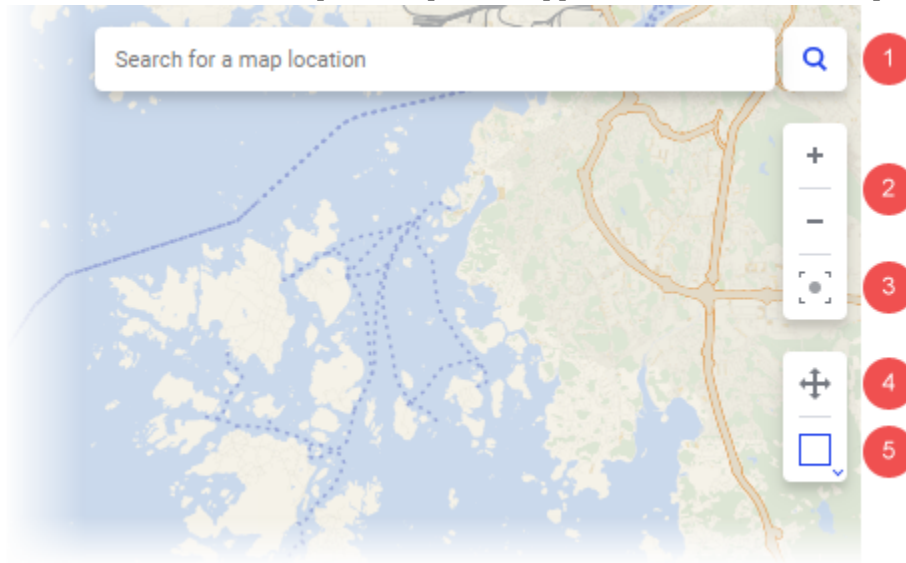
- On the map chart, right-click the position you want to view the coordinates for. For a map, where a coordinate reference system is specified, the latitude and longitude for the position are displayed to the right of **Copy coordinates**. For a map, where the coordinate reference systems are set to None, the Cartesian coordinates of the position are displayed to the right of **Copy coordinates**.



You can copy the values by clicking the **Copy coordinates** item in the menu, and then paste them into a document.

## Navigation in the map chart

The zooming and interaction controls are located at the top right of the visualization. To select which controls to show in the map chart, open the Appearance section of the Map chart Properties.



### 1. Search

Use the search field to find locations on the map. Click the magnifying glass to show or hide the search field.

### 2. Zoom

Click the plus sign to zoom in, and the minus sign to zoom out. You can also use the mouse wheel to zoom in or out.



Select **Scroll page instead of zooming map** in the Appearance section of the Map chart properties if you want to use the mouse wheel primarily to scroll the analysis page rather than zooming in and out on the map. You can still zoom using the mouse wheel, just press the Ctrl key and scroll simultaneously.

### 3. Zoom to filtered items

Click to zoom in or out to the filtered values in the map chart.



If you want to automatically zoom to filtered markers or shapes as you filter, right-click in the map chart and select **Auto-zoom**.


### 4. Panning mode


When panning mode is selected, you can sweep the map in different directions by clicking and dragging. Double-click on the map to zoom in one step. You can still mark items in panning mode by clicking a single item or by pressing Ctrl and clicking multiple items.

### 5. Marking mode

When marking mode is selected, you can mark multiple items in the map chart, using one of the marking modes. Click to open the menu, and select the marking mode you prefer.



 Rectangle marking – Click and drag the mouse pointer to draw a rectangle enclosing the interesting items.

 Lasso marking – Click and drag the mouse pointer to draw an area of any shape around the interesting items.

 Radius marking – Click and drag the mouse pointer to draw a circle enclosing the interesting items.

When you use the default map projection (EPSG:3587 - WGS 84/PseudoMercator), the length and width of the rectangle are displayed when using rectangle marking, and the radius distance when using radius marking. For other map projections, distances will not be visible.

## Shortcuts

The table below describes some keyboard shortcuts that you can use when the map is in marking mode.

Shortcut	Description
Press and hold the Ctrl key and the right mouse button simultaneously to grab the map and move it in any direction.	Pan in any direction when map chart is in marking mode.
Press and hold the Shift key and the left mouse button simultaneously, while drawing a circle with the mouse pointer.	Temporarily switch to radius marking when another marking mode is selected.
Press and hold the Alt key and the left mouse button simultaneously, while drawing a custom area with the mouse pointer.	Temporarily switch to lasso marking when another marking mode is selected.
Press and hold the Shift + Ctrl + Alt keys and the left mouse button simultaneously, while drawing a rectangle with the mouse pointer.	Zoom in on a rectangular area of the map chart when rectangular marking mode is selected.

## Fixing geocoding issues

Sometimes the automatic geocoding does not work as expected. Below are some tips of what you can do to fix some geocoding issues.

For general information, see [Geographic location and geocoding](#) on page 169.

### Column titles not recognized as geographical titles

The title of a column does not always indicate that the content is geographical.

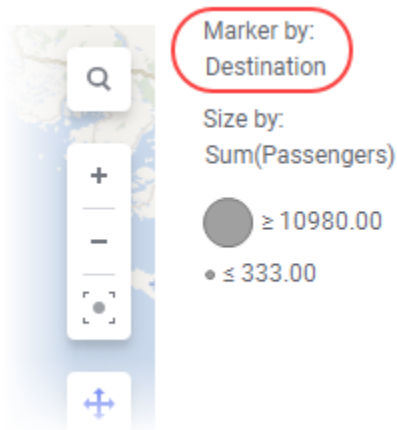
Assume you want to position markers at the locations below in a marker layer.

Destination	Passengers
Rome	10980
Stockholm	560
Prague	741
Cairo	1098
New Delhi	8009
Madrid	450
Casablanca	333


Simply creating a layer with markers will not show what you expect; no markers are shown. An exclamation mark in the map chart title signals that your map chart has not been configured. (You can click the exclamation mark to get more information.) This is because the title "Destination" is not recognized as a column containing geographical locations. Consequently it is not automatically matched against the geocoding tables.



Simply specify the column to be matched, that is "Destination", by selecting it in the **Marker by** selector in the marker layer legend (or in the Properties), and the markers get positioned correctly.



### Matches not found in the geocoding tables

The geographical names in your data may not always match the names in the geocoding tables exactly. The symbol  in the map chart title alerts you. Clicking the symbol lists the five first mismatches found, but more may exist.

Check whether mismatches depend on spelling mistakes in your data table. If that is the case, it might be fixed by [replacing values](#).

Also be aware of that a geographical name in a geocoding table may be expressed in a different language than what is used in your data. For example, in geocoding tables covering the world, names may be expressed in the official language used in each country.

### Same geographic name at several locations

Geographical names are sometimes identical but located at different places. The exclamation mark informs you of the multiple entries. If appropriate columns are available, avoid the issue by specifying a geographical hierarchy, for example, Country > State > City.

### Turn off geocoding warnings when the map chart is in a good shape

If geocoding warnings are displayed in your map chart, and you deem the warnings to be acceptable, you can turn them off to avoid that the map chart looks broken when it is not. Right-click the visualization, then open the **Appearance** section for the map chart, and clear the check box named **Show geocoding warnings** to hide the warnings from the visualization title.

## Aggregating coordinate-based locations values

If geographic coordinates are available in the data loaded into a marker layer in the map chart, each marker is positioned at its coordinates. But you might not be interested in separate markers; instead an aggregated value representing multiple markers might be of interest.

The most common way to position markers is by [geocoding](#). Then you can aggregate values by moving up one level in the specified geographic hierarchy. For example, if the columns you use for geocoding are Country > City, you simply remove the "City" column to get a marker representing the aggregated value for all the cities in each country.

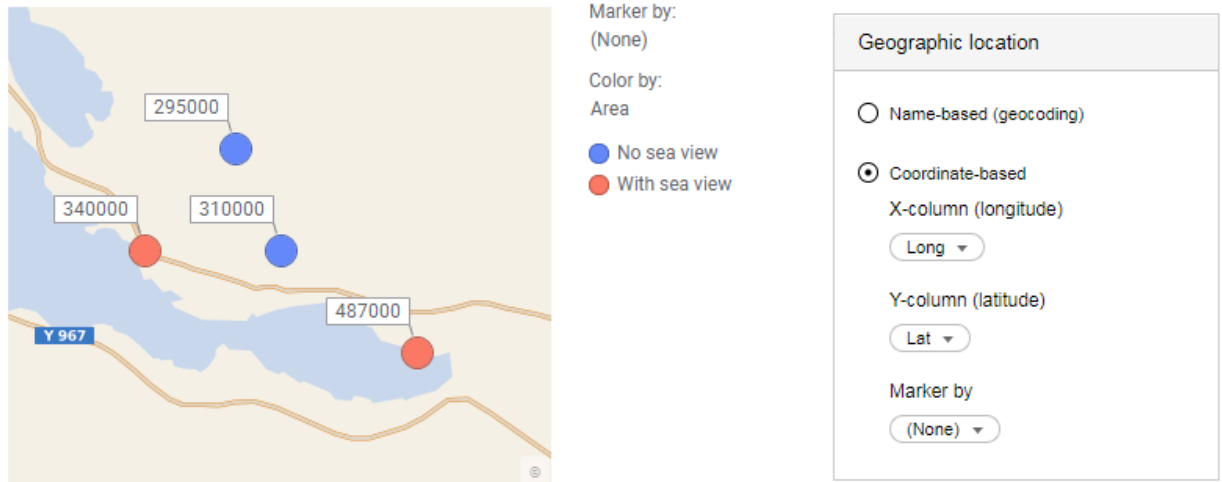
If you want to aggregate values for markers positioned using coordinates, first check whether geocoding can be used, that is, switch to **Name-based (geocoding)** in the Properties, the **Geographic location** section. However, this option requires that columns with geographical names are provided. If no such columns are provided, see the example of a possible solution below.

### Example

The data contains house prices for four different objects. Two of the objects are located by the sea, and two of them are further into the land. Note that no columns with geographical names are available, only coordinates.

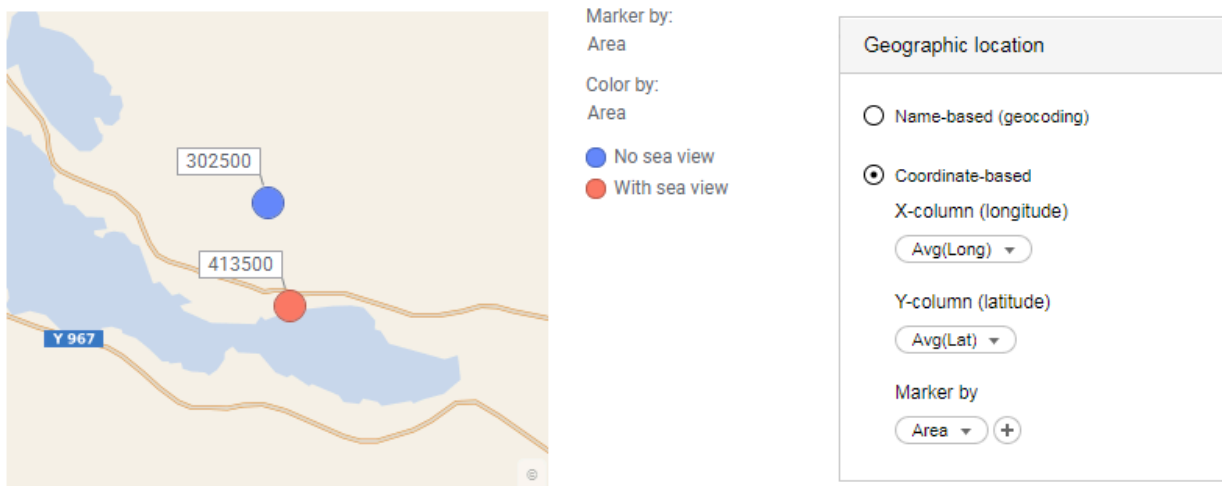
Object ID	Area	Price	Lat	Long
1	With sea view	340000	63.54	17.29
2	No sea view	295000	63.55	17.31
3	With sea view	487000	63.53	17.35
4	No sea view	310000	63.54	17.32

Below, the data is loaded into a map with the objects placed at their coordinates (labels show the prices):



Assume you wish to compare the average price of houses with a sea view, to houses without. Geocoding is not an option, since no column contains geographic names.

Instead, select "Area" on the **Marker by** selector to aggregate values per "Area" type. Also the X- and Y-column values must be aggregated. Otherwise each of the four coordinates will be displayed. The best aggregation method is probably average as shown in the **Geographic location** section below:



Note the labels have been specified to show the average price. You specify labels in the **Labels** section in the Properties.

## Example of a line connection in a marker layer

Lines can be drawn in a map layer with markers to show connections between specific markers. You can specify in which order the markers should be connected, and arrows can be added to indicate the direction.

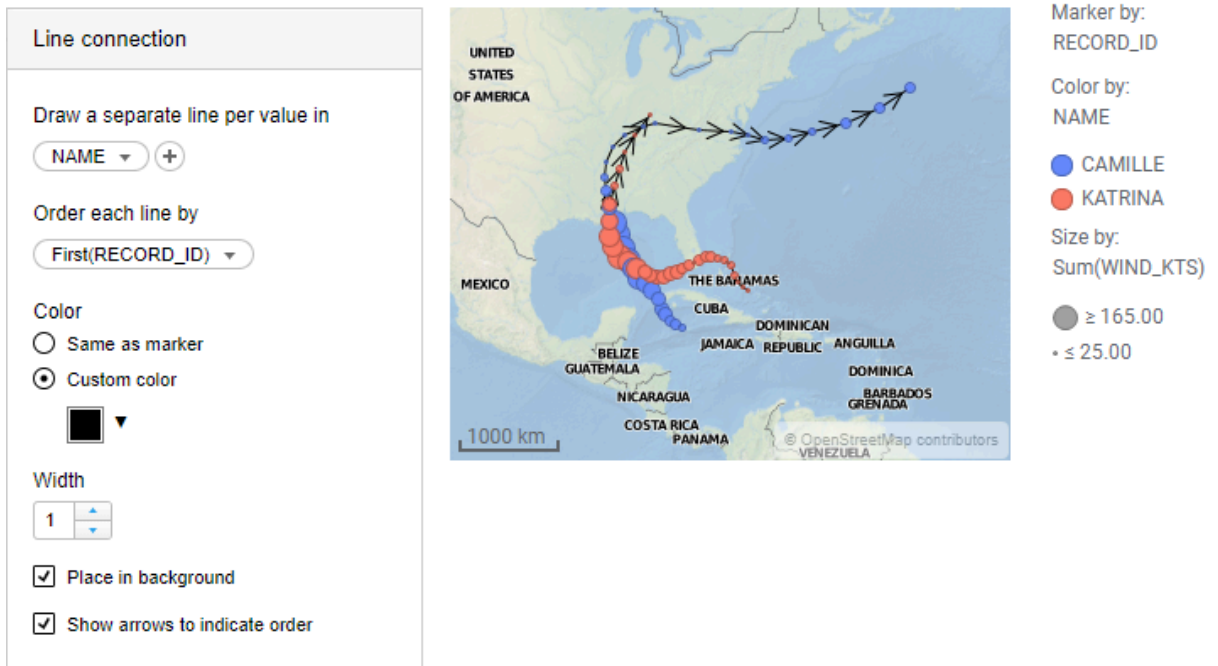
The data table below contains wind force measures at certain "LAT" and "LONG" coordinates during the advance of two hurricanes, Camille and Katrina. The "RECORD\_ID" column values keep track of the chronological order of the measures.

NAME	LAT	LONG	RECORD_ID	WIND_KTS
CAMILLE	37.30	-68.40	24882	50
CAMILLE	38.00	-64.90	24883	55
CAMILLE	39.20	-61.40	24884	60
CAMILLE	40.80	-58.20	24885	55
KATRINA	23.10	-75.10	36714	30
KATRINA	23.40	-75.70	36715	30
KATRINA	23.80	-76.20	36716	30
KATRINA	24.50	-76.50	36717	35
KATRINA	25.40	-76.90	36718	40

Creation of a marker layer in the map chart will position markers at each of the provided coordinates:



To emphasize the hurricanes' advance, you can connect the markers with lines. The settings are made in the Line connection section in the marker layer Properties:

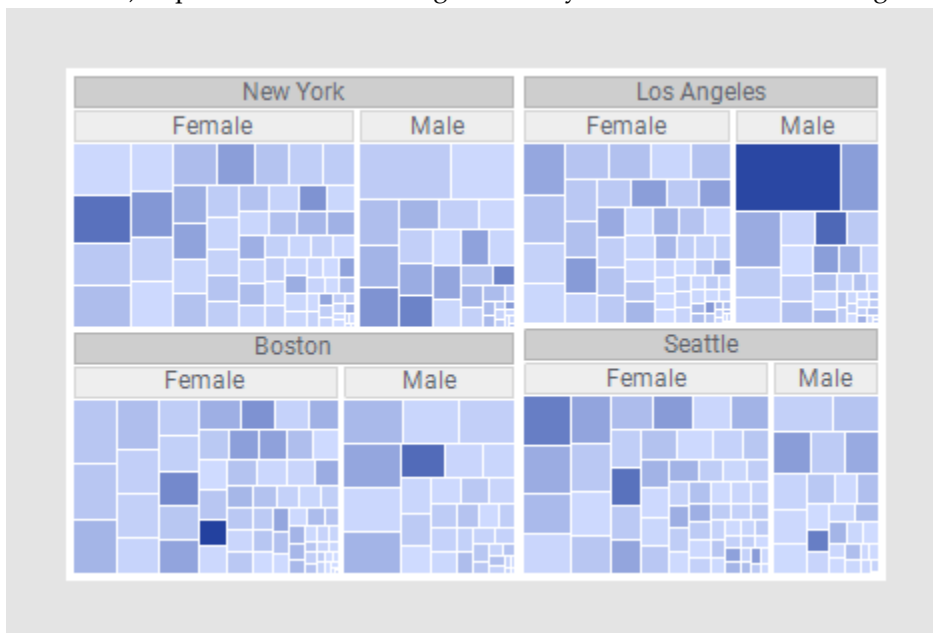


Separate lines are drawn for the two hurricanes as specified in the topmost setting. The "RECORDED\_ID" column, that lists the order of measures, is selected below **Order each line by** to indicate in which order the markers should be connected. The added lines result in curves that display how each hurricane moves.

As shown, line connections can be used to visualize movements. Another use of line connections could be displaying routes between destinations.

## Treemap

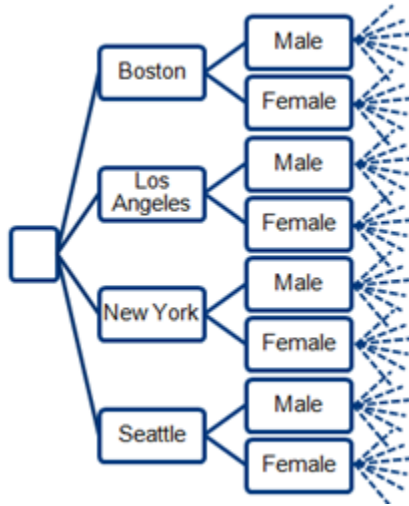
A treemap is used for displaying huge amounts of data that can be structured hierarchically (tree-structured). It presents the data using differently sized and colored rectangles.



Data in a treemap is organized hierarchically. Each branch in the hierarchy is symbolized by a rectangle, which in turn is divided into smaller rectangles representing sub-branches. Within the

different hierarchy levels, the rectangles are sized proportionally by the values in a numerical data column, for example a sales column. The largest rectangle is positioned in the top left corner and the smallest rectangle in the bottom right corner.

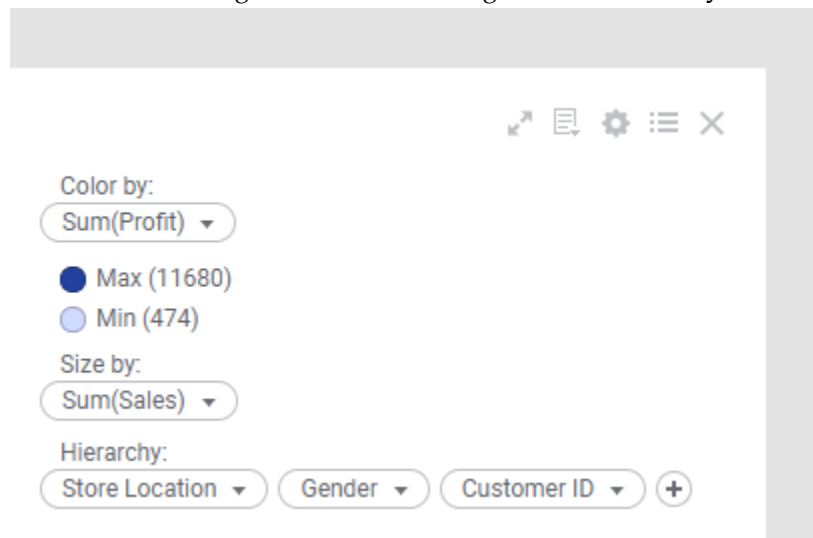
For example, below you can see the hierarchy used in the treemap above:



The entire visualization space is used, which makes it possible to present a large amount of data simultaneously. This makes it easy to spot patterns and irregularities in your data by a quick look.

The sizes of the rectangles represent numerical data. Also the colors of the rectangles can reflect numerical data by changing gradually between a color representing the maximum value of your data and another color representing the minimum value. You can choose the same data column for both sizing and coloring, or different data columns. In the treemap above, the sizes of the rectangles reflect one data column, and the colors another.

The different settings are found in the legend: the **Hierarchy** axis, the **Size** axis, and the **Color** axis.




## Creating a treemap

A treemap is used for displaying huge amounts of data that can be structured hierarchically (tree-structured). It presents the data using differently sized and colored rectangles.

### Prerequisites

See [Treemap](#) on page 424 for a description of the concept.

### Procedure

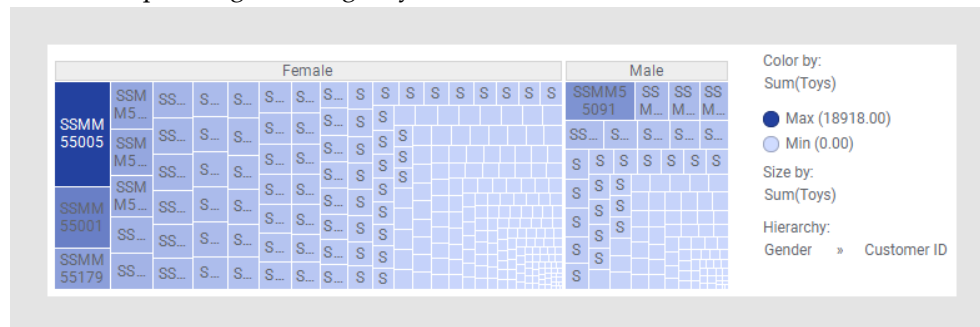
1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Treemap** visualization type to the wanted position on the analysis page.  
A suggestion of a treemap is presented.
3. On the **Hierarchy** axis, [select the columns](#) to include in the hierarchy. Start with the top level.
4. On the **Size** axis, [select by which column](#) the rectangles should be sized.
5. [Select which type of aggregated value](#) to display for the selected column.
6. On the **Color** axis, [select by which column](#) the rectangles should be colored.
7. [Select which type of aggregated value](#) to display for the selected column.

### Example

The data table below lists how much money different customers have spent at four different store departments. Assume you want to find the "top" male and female customer at the Toys department.

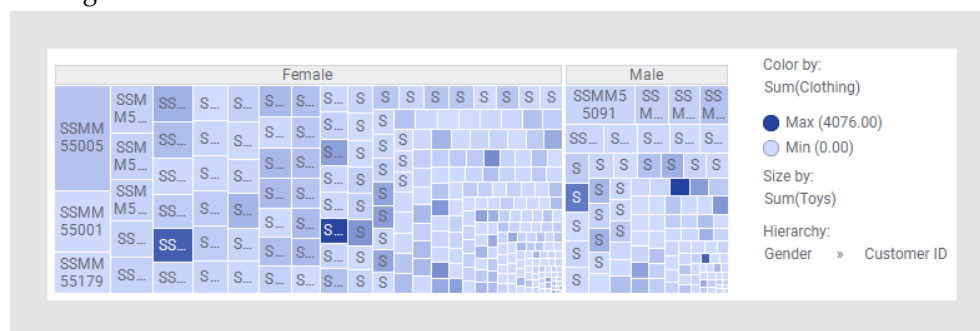
Customer ID	Gender	Electronics	Clothing	Furniture	Toys
SSMM55001	Female	10013	19	3488	11266
SSMM55002	Female	6396	952	5774	185
SSMM55003	Female	5586	342	1369	354
SSMM55004	Male	885	363	1362	0
SSMM55005	Female	4860	626	5865	18918
SSMM55006	Female	216	1276	524	304
SSMM55007	Female	436	10	0	0
SSMM55008	Female	0	0	0	0

The treemap settings below give you the answer.



Customer ID SSMM55005 is the woman who spent most money, and SSMM55091 is the man, because these customers' rectangles are the largest in the female and male branches respectively.

Maybe these two customers are the "top" customers at the Clothing department as well. To find out, change the column on the **Color** axis to Clothing to see if these rectangles have the darkest color.



It seems other customers are more interested in clothing than the two customers who spent the most on toys!

## Navigating in a treemap hierarchy

You can navigate downwards as well as upwards in the treemap hierarchy.

Navigating downwards in the treemap hierarchy makes it possible to view the details of a level in a branch. The branch level in question fills the entire visualization, and other branches are no longer visible. After viewing the details, you can navigate upwards again.

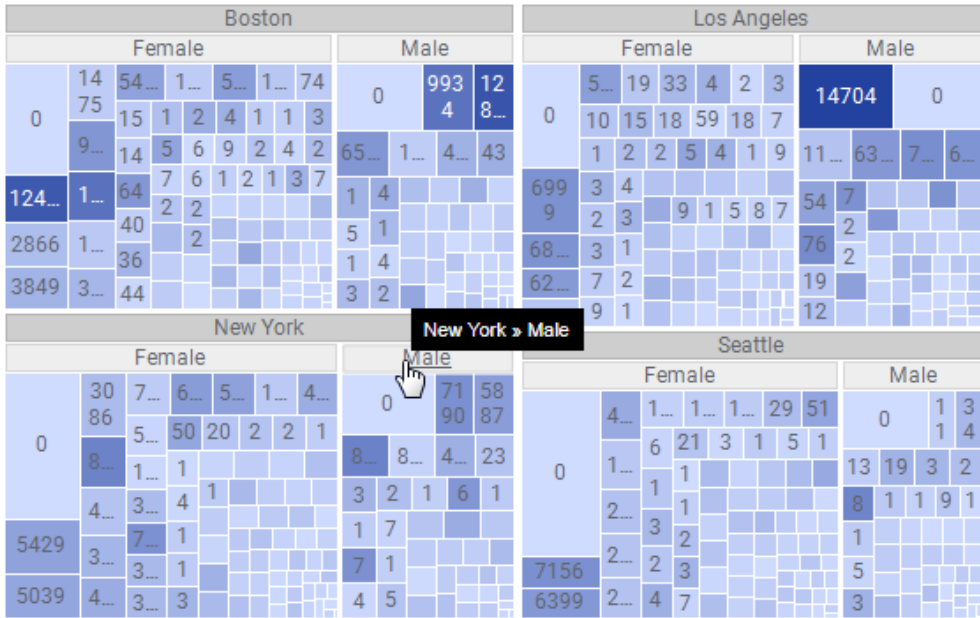
### Procedure

1. In the treemap, move the cursor over the header of the branch level you want to view the details for. The header becomes underlined.
2. Click the header.  
The branch level in question fills the entire visualization. At the uppermost header of the treemap, the path of the hierarchy levels is displayed.
3. To return to an overview again, that is, navigate upwards in the hierarchy, click the wanted level in the hierarchy path.

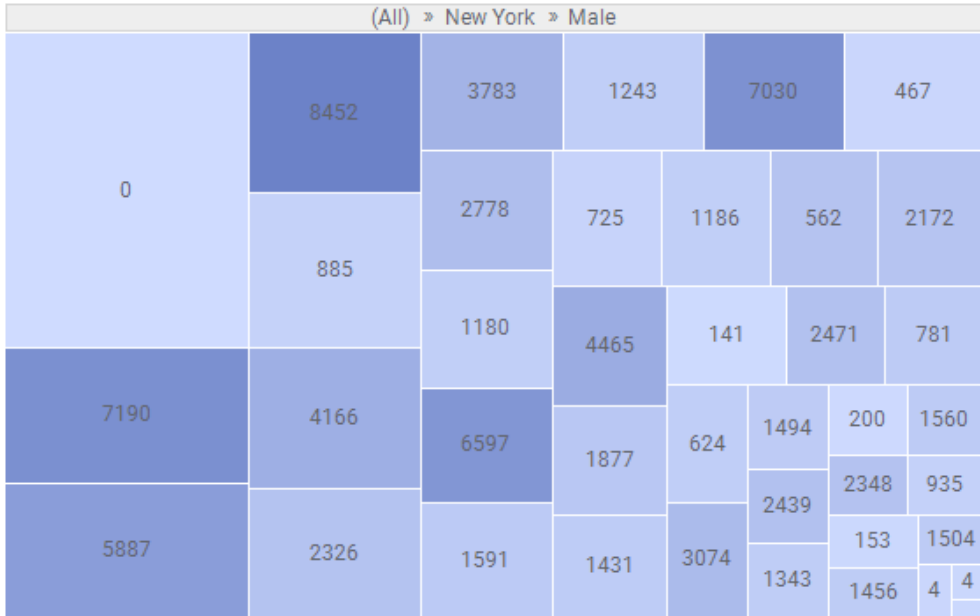


# Example

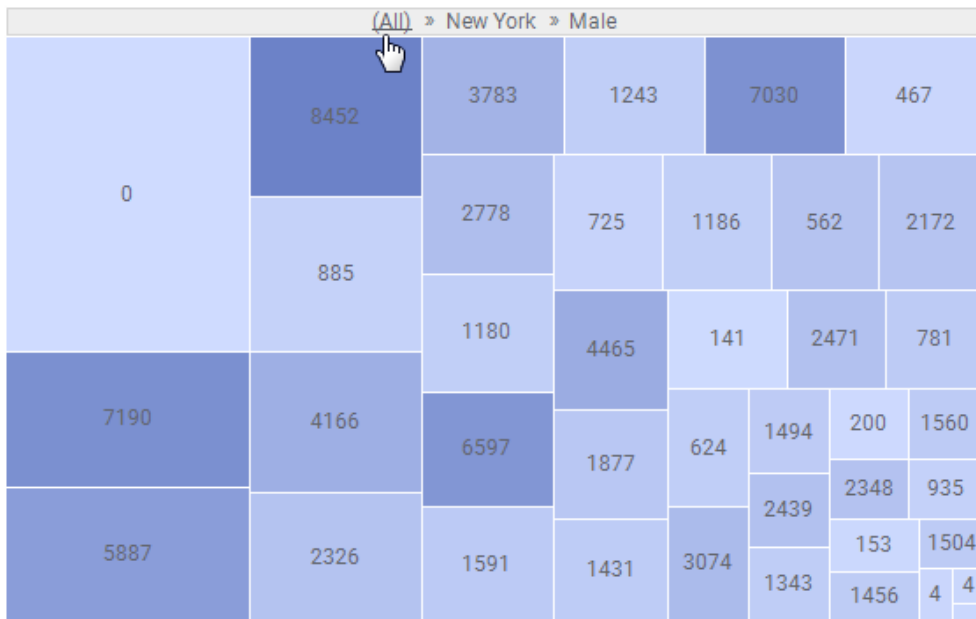
If you want to take a closer look at the 'New York >> Male' level, click the 'Male' header as shown below.



In the resulting treemap, you can now view more details:



To navigate upwards in the hierarchy, click (All) in the hierarchy path to return to the initial view. Alternatively, you can move only one step up in the hierarchy by clicking 'New York' instead.



## Heat map

The easiest way to understand a heat map is to think of a cross table or spreadsheet, in which numerical values are represented by colored cells instead of the values themselves. The default color gradient sets the lowest value in the heat map to dark blue, the highest value to a bright red, and mid-range values to light gray, with a corresponding transition (or gradient) between these extremes.

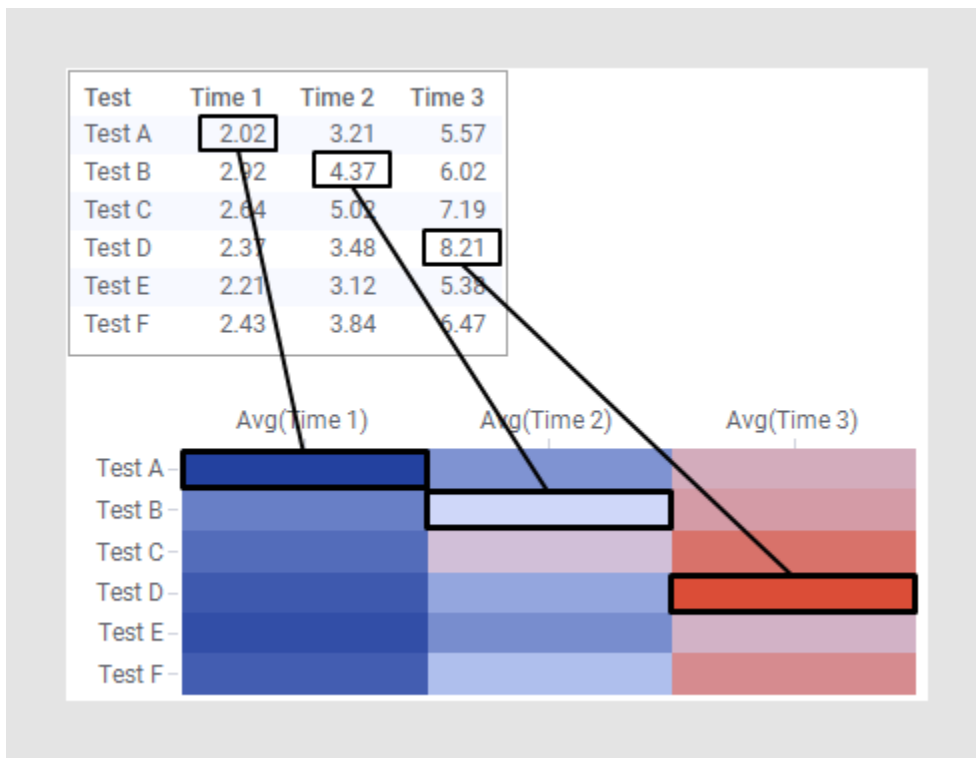


Heat maps must be authored in the installed client.

Heat maps are well-suited for visualizing large amounts of multi-dimensional data and can be used to identify clusters of rows with similar values, as these are displayed as areas of similar color.

### Example

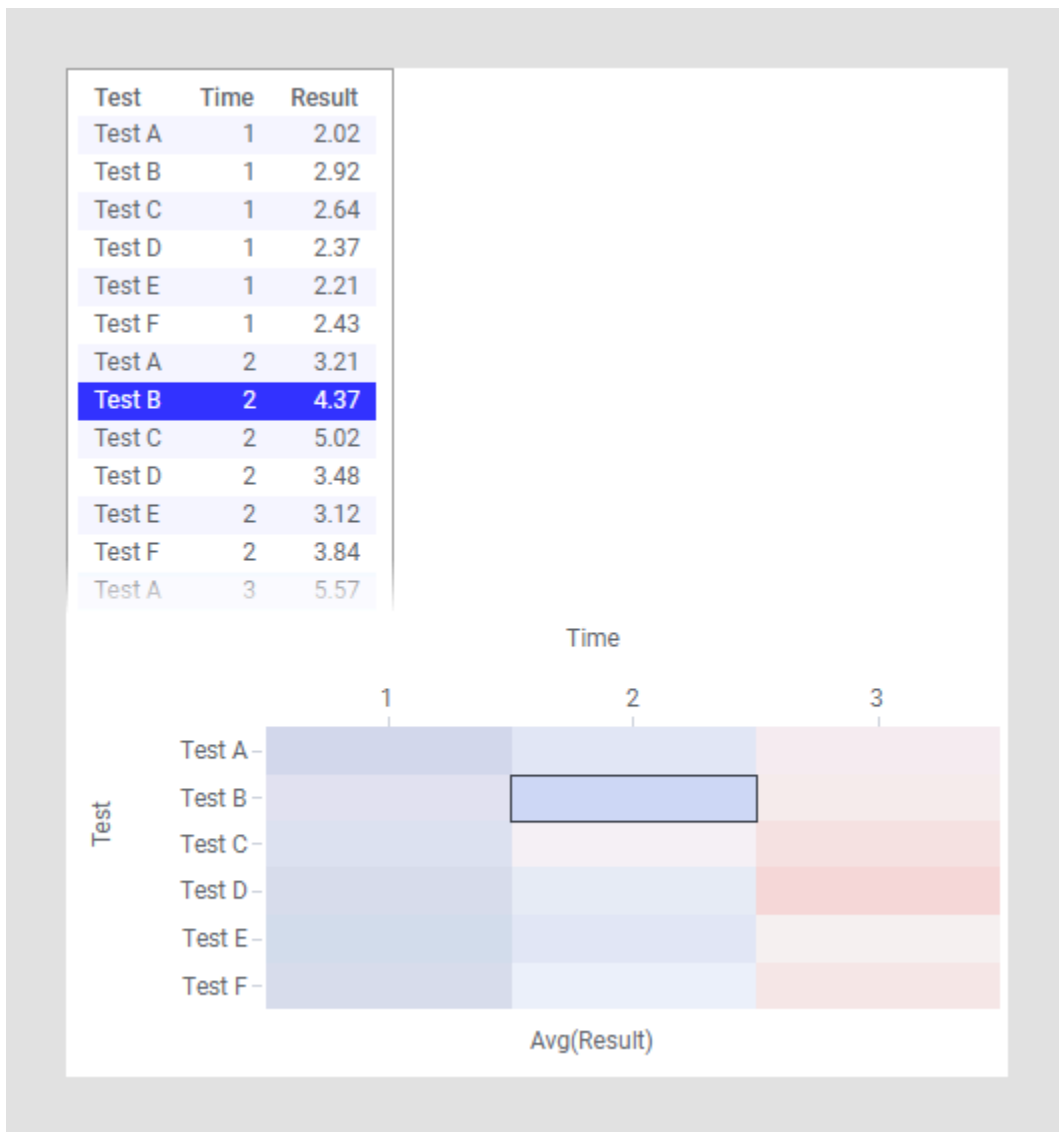
The example below shows how the values in the table are displayed as color gradients in the heat map cells.



As the examples below will illustrate; how to configure a heat map depends on the format of your data.

### Data in tall/skinny format

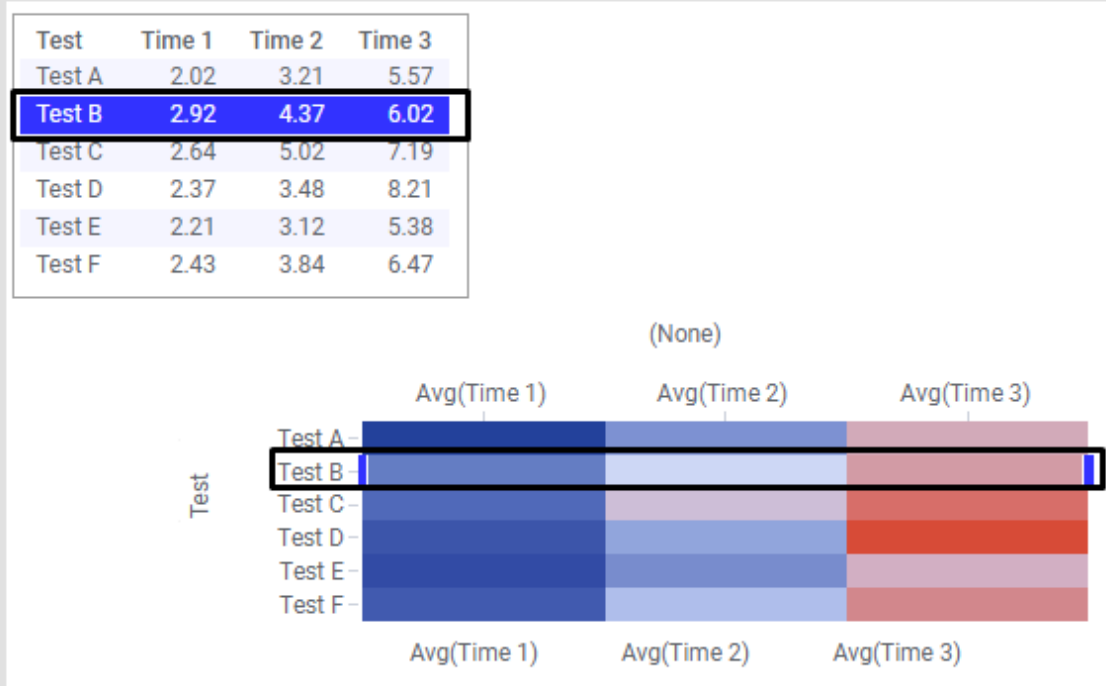
In this example, the data is in tall/skinny format, and each data table row corresponds to a single cell in the heat map.



The column Test is selected on the Y-axis, Time on the X-axis, and the cell values are set to the column Result. Like in other visualizations, highlighting and marking in the heat map are applied to one or more rows in the underlying data table. This means that with the configuration in this example, marking a row in the table will mark a cell in the heat map.

### Data in short/wide format

In this example, the data is in short/wide format instead, and each row in the data table corresponds to an entire row in the heat map.



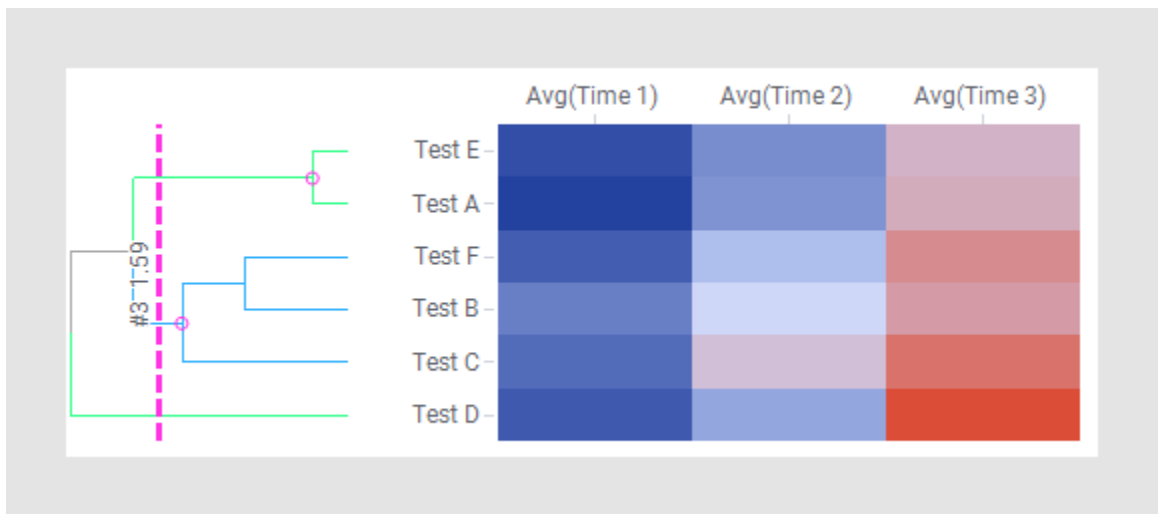
The Y-axis in this heat map is set up with the column Test, while the X-axis is set to (None). For the individual cell values in the heat map, the columns Time 1, Time 2, and Time 3 are selected. Cell value columns are always aggregated unless the Y-axis is set to (Row Number). This is because the data table may contain many rows with the same name, and the values in these rows must then be aggregated to one single value to be displayed in the heat map. The content of the data is the same as in the example with data in tall/skinny format, but the format of the data makes it necessary to configure the heat map in a different way. With data in short/wide format, this is a common way to configure a heat map.

## Dendrograms

It is often useful to combine heat maps with hierarchical clustering, which is a way of arranging items in a hierarchy based on the distance or similarity between them. The result of a clustering calculation is presented either as the distance or the similarity between the clustered items depending on the selected distance measure.

You can cluster both rows and columns in the heat map. The result of a hierarchical clustering calculation is displayed in a heat map as a dendrogram, which is a tree-structure of the hierarchy. Row dendrograms show the distance (or similarity) between rows and which node each row belongs to is a result of the clustering calculation. Column dendrograms show the distance (or similarity) between the variables (the selected cell value columns).

The example below shows a heat map with a row dendrogram where the distance between the rows were calculated.

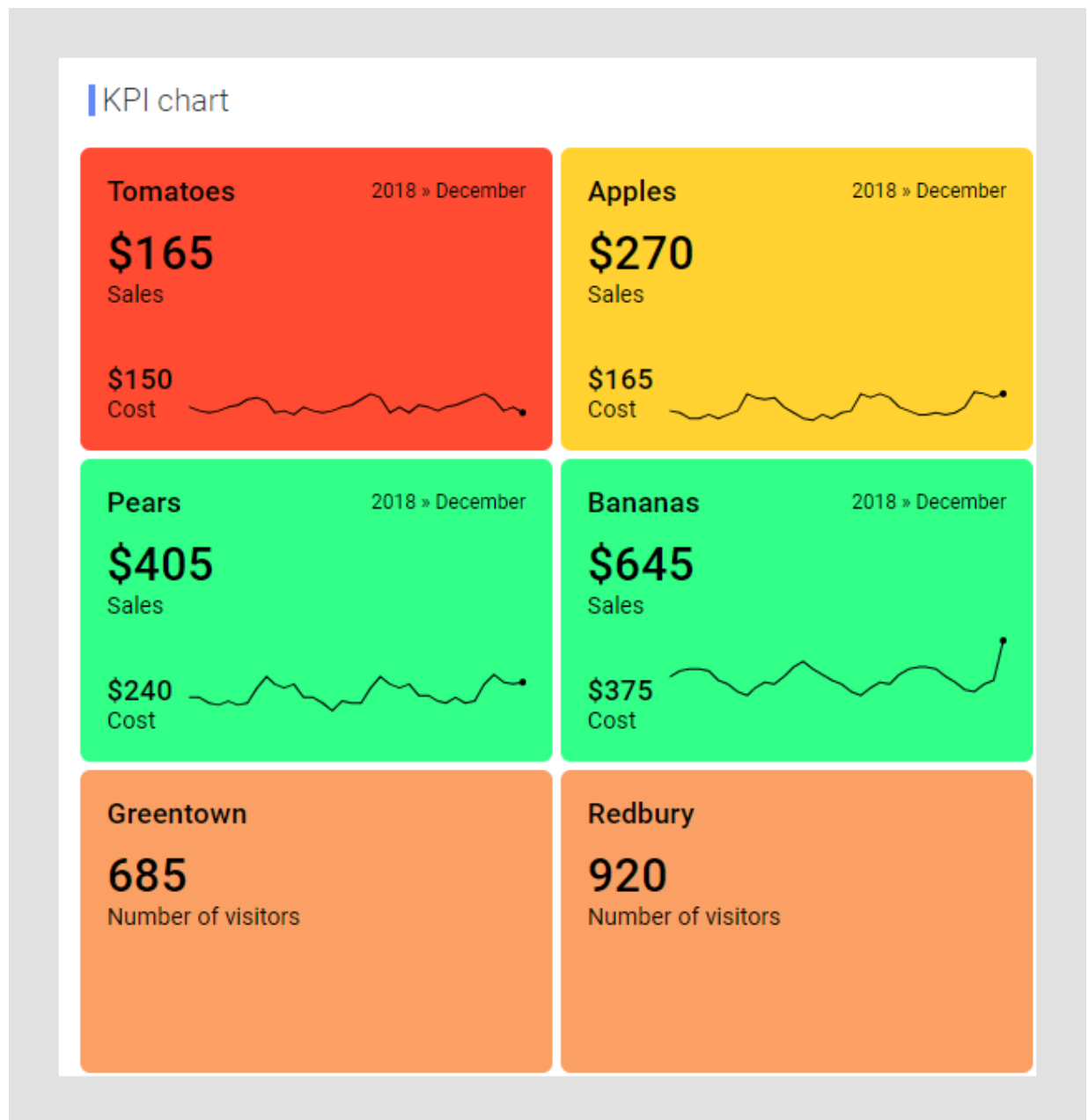


As a result of the clustering calculation, the rows in the heat map have been reordered to correspond to the cluster calculation. Test A and Test E are placed in the same cluster. Test F and Test B are placed together in another cluster, and this cluster forms another cluster together with Test C. Test D is not included in any of those clusters. This indicates that Test A and Test E are closer to each other than what they are to Test F, Test B, Test C, or Test D. It also indicates that Test D is the one that is the most distant to any of the other rows.

## KPI chart

A KPI chart is used to quickly inform about the current performance of a company or organization. You create different KPIs, Key Performance Indicators, which measure factors that are crucial to monitor,

and present them in a grid of tiles. Examples of KPIs to monitor are net revenue, sales growth, or customer satisfaction.



A single KPI chart can hold KPIs showing different types of measures. In the image above, one KPI shows sales and cost data and another KPI shows data about visitors. Each individual KPI can be split into tiles that represent different categories in your data.

You have different options to arrange the tiles from the different KPIs in a single KPI chart. You can, for example, [gather tiles where the KPIs show good performance at the top](#) of the chart, or you can [sort them alphabetically](#). That is, tiles from different KPIs can be mixed, and their values can be based on data from different data tables.

Performance can be visualized in different ways in a tile. The tile details are described in [Adding a new KPI to a KPI chart](#).



**Important** The values in the tiles show the performance for the most recent time period. This time period can be displayed in the upper right corner of the tiles.


## Creating a KPI chart

A KPI chart is used to quickly inform about the current performance of a company or organization. One or more KPIs, Key Performance Indicators, are presented in a grid of tiles.

### Prerequisites

See [KPI chart](#) on page 434 for a description of the concept.

### Procedure

1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **KPI chart** visualization type to the wanted position on the analysis page.  
A suggestion of a KPI chart is presented.
3. Adjust the KPI to display the measures you want to monitor. See [Adding a new KPI to a KPI chart](#) for the details on how to specify the different measures.



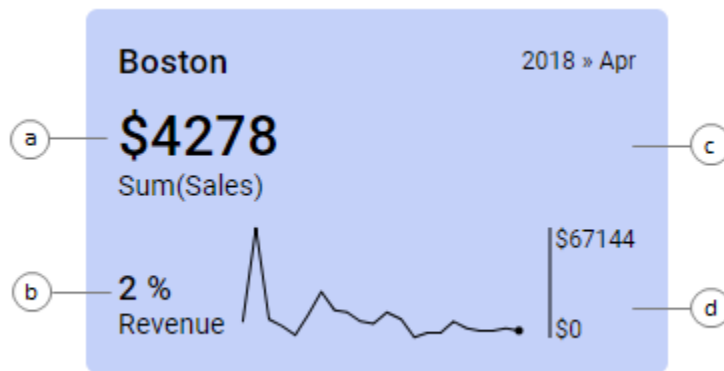
You can add more than one KPI to the KPI chart. Each KPI has its own properties to be specified. There are also [properties that are generic for the entire KPI chart](#).

4. If you want, you can further adjust the layout of your KPI chart. You can specify to start each KPI on a new row, the minimum tile width or how many tiles should at least be visible.

## Adding a new KPI to a KPI chart

A single KPI chart can display several KPIs. Which performance values that are displayed in the KPI tiles can differ from KPI to KPI. When adding a new KPI to a KPI chart, you define which values to monitor in the tiles, and for which categories.

The values of a KPI are displayed in tiles, where each tile represents a certain category (Boston in this example) .



There are different ways to visualize these performance values as illustrated above. You can

- display a primary value, which shows the most recent measure of a certain factor (a). What is measured is displayed beneath the value.
- display a comparative value, to which the primary value is compared or evaluated (b). Examples are comparisons of the actual value to a target, or to a corresponding value for another time period. What is measured is also displayed here.
- use colors of the tile backgrounds to indicate the current performance level (c). Rules for the coloring can be configured. They can, for example, contain threshold values against which the current primary value is valuated. Depending on the results of the evaluation, the tile backgrounds are colored differently.



- display a sparkline that shows the primary value over time (d). The dot ending a sparkline indicates the current value. The sparklines can be supplemented with a scale showing the vertical range.

### Prerequisites

A [KPI chart is created](#).

### Procedure

- Right-click a KPI in the KPI chart, and select **Properties** in the opened menu.  
The Properties pop-over for the selected KPI is displayed. In the side panel, the current KPI is highlighted.
- At the bottom of the side panel, click **Add new KPI**.  
The new KPI is added to the chart. In the side panel, the new KPI is highlighted.
- In the pop-over, select **Values**.  
The Values section is shown.
- On **Value (y-axis)**, specify what to be calculated and displayed as the primary value in the tiles. You can
  - [select a column](#) and an [aggregation](#) in the column selector.
  - right-click the column selector and click **Custom expression**. This opens a dialog where you can type your own expression.

The result of the calculation for the most recent time period is displayed as the primary value in the tiles. For time period, see the Time setting in the next step.
- On **Time (x-axis)**, select the column or hierarchy to use for defining the time periods.  
The displayed primary value is the result for the most recent time period of the periods you have defined.
 

To show the time period in the tile, select the **Show time in tile** check box beneath the **Time** axis. To indicate the development of the primary value over all the time periods, sparklines are used.
- Under **Tile by**, select the column that specifies which categories the tiles should represent.  
The categories are displayed at the upper left of the tiles.
- On **Comparative value**, use any of the options described in Step 4 to specify what the primary values should be compared to.  
The result for the most recent time period is displayed as the comparative value in the tiles.
- If you want to use colors for indicating performance, select **Colors** in the pop-over.  
The Colors section is shown. The default colors of the tile backgrounds reflect the primary values, and a gradient color transition is used.
- On **Color by**, use any of the options described in Step 4 to specify what calculation the coloring of the tile backgrounds should be based on.
- To let background colors of the tiles indicate performance levels in a different way, click **Add color rule**.  
The Add color rule dialog is displayed.
- In the dialog, [add color rules](#) that control how visualization items should be colored.

### Result

The KPI is added to the KPI chart.



Select the check box **Start each KPI on a new row** to be able to better distinguish the individual KPIs from each other and to make the whole visualization clearer.

## Basic example of a KPI

The principles for creating a KPI, key performance indicator, are explained using a simple example.

For details about the different settings in a KPI, see [Adding a new KPI to a KPI chart](#).

The data table below lists, for four days, the purchase and sales figures for some fruit types.

Fruit	Sales	Purchase	Date
Apples	70	70	3/26/2018
Bananas	90	100	3/26/2018
Oranges	60	50	3/26/2018
Pears	100	70	3/26/2018
Apples	280	60	3/27/2018
Bananas	70	60	3/27/2018
Oranges	60	70	3/27/2018
Pears	220	100	3/27/2018
Apples	100	75	3/28/2018
Bananas	150	100	3/28/2018
Oranges	300	260	3/28/2018
Pears	120	80	3/28/2018
Apples	250	200	3/29/2018
Bananas	160	80	3/29/2018
Oranges	270	180	3/29/2018
Pears	80	50	3/29/2018

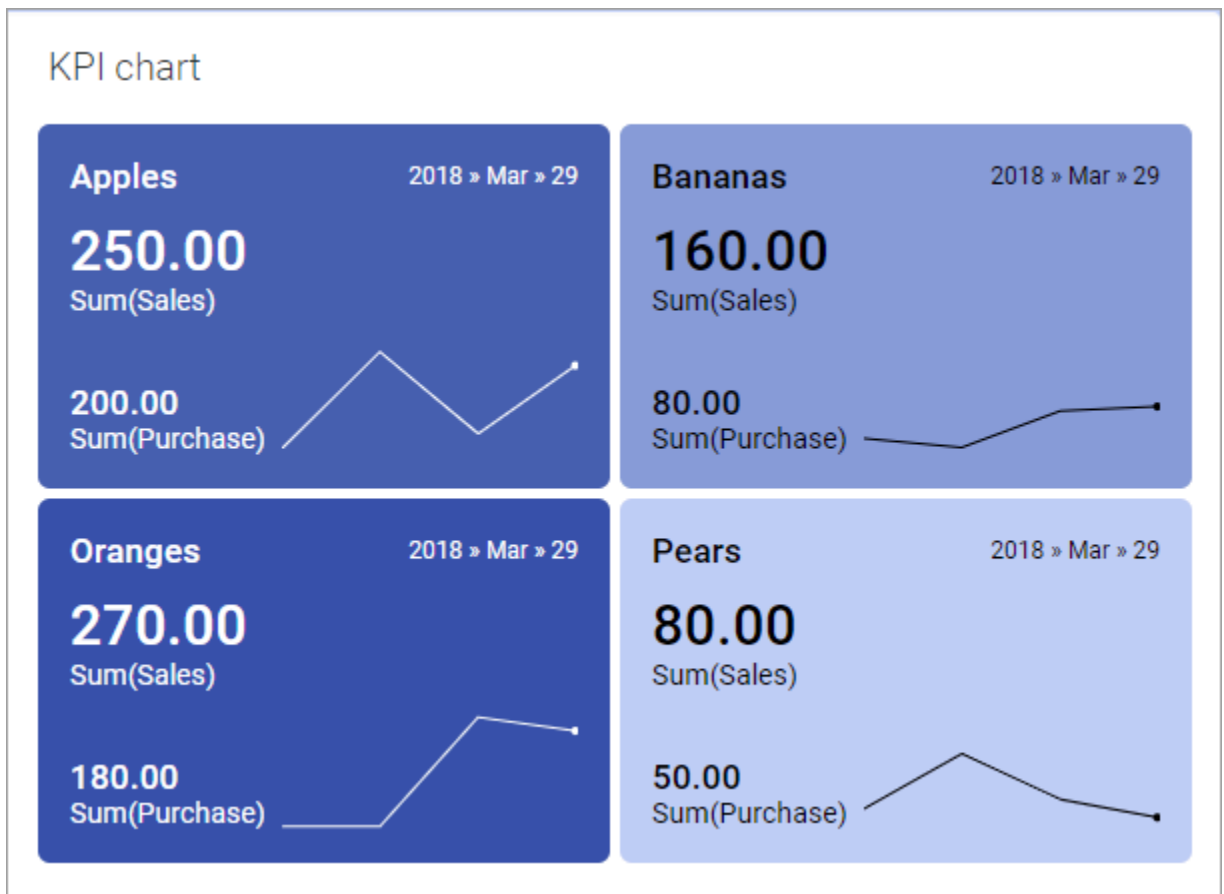
Assume you want to monitor the latest sales performance regarding the absolute figures for sales and purchases per day and fruit type. You also want to be alerted on sales figures less than 200 per day.

Creating a KPI chart from scratch based on the data above results in this KPI:

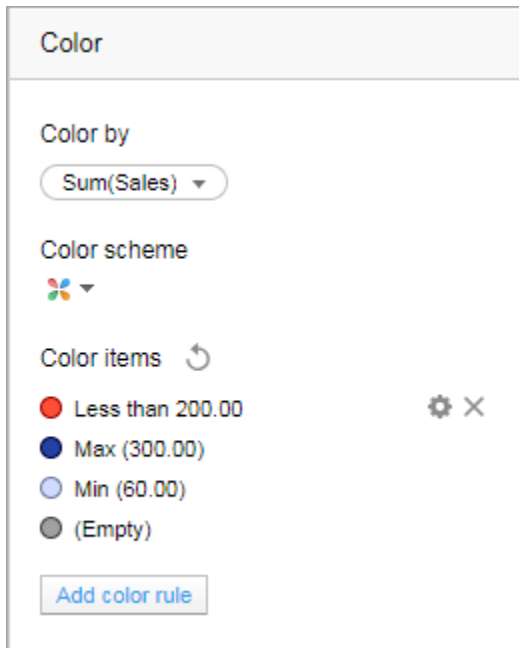


The Sales figures for the most recent date are shown as the [primary values](#) in the tiles. A sparkline, ended by a dot indicating the current value, shows the sales trend over the four days.

To keep an eye of also the Purchase figures, you can let the [comparative values](#) in the tiles show these figures. The image below shows the result. In addition, the **Show time in tile** check box is selected to indicate the time periods in question.



The background colors of the tiles can be used to indicate performance levels. You can for example distinguish tiles with sales figures less than 200 by using a red color. In the Color section of the Properties pop-over, a [color rule is added](#) that colors tiles with this condition fulfilled in red:

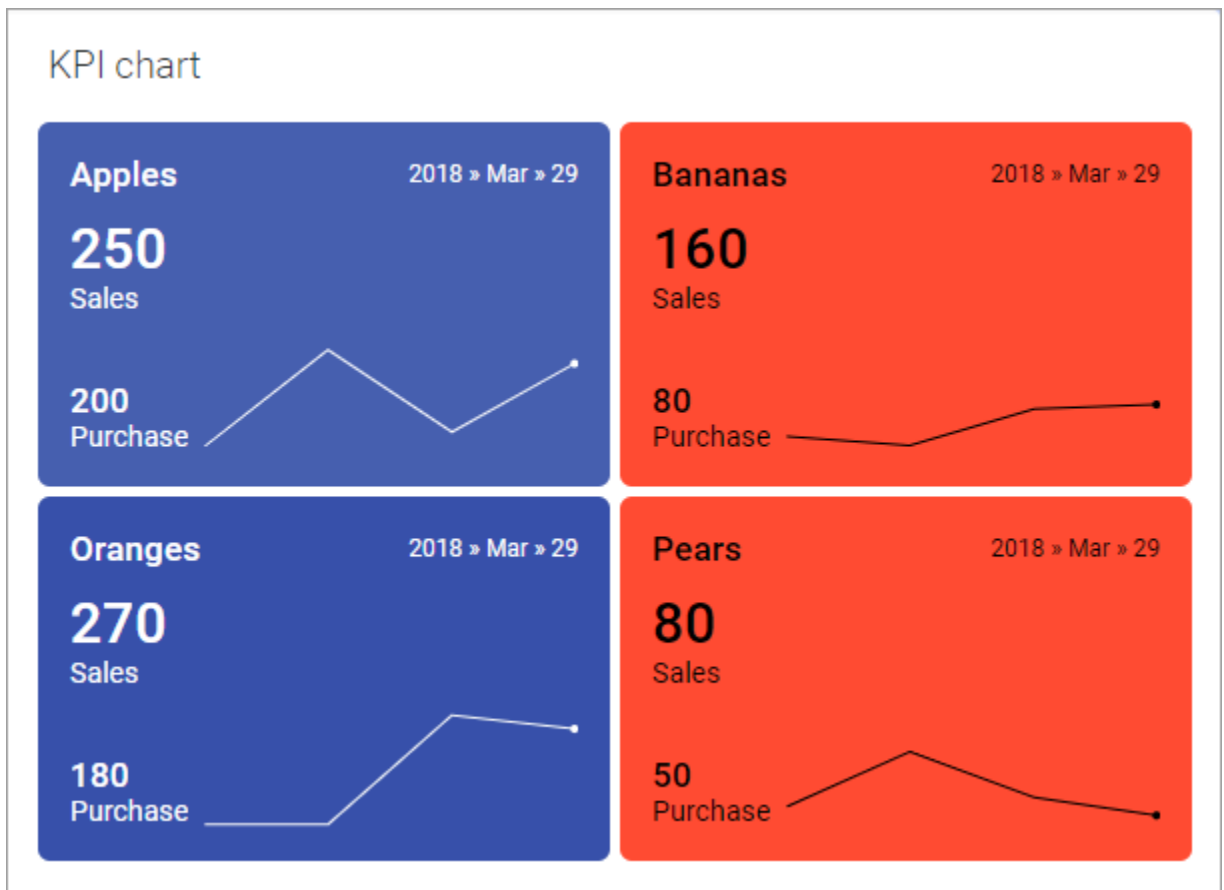


The resulting KPI shown below.

## KPI chart



Furthermore, you can refine the appearance by [changing the display name of an axis](#) and [changing the formatting of a column](#).



### KPI showing absolute and relative profit

The following example gives a hint of how to create a KPI that shows absolute and relative profit in the tiles. The starting point is a data table that contains columns with sales and purchase figures for a number of products. To retrieve these KPI values, custom expressions are used on the KPI axes.

For details about specifying different KPI values that are mentioned in the example, see [Adding a new KPI to a KPI chart](#).

The data table below lists, for four days, the purchase and sales figures for some fruit types.

Fruit	Sales	Purchase	Date
Apples	70	70	3/26/2018
Bananas	90	100	3/26/2018
Oranges	60	50	3/26/2018
Pears	100	70	3/26/2018
Apples	280	60	3/27/2018
Bananas	70	60	3/27/2018
Oranges	60	70	3/27/2018
Pears	220	100	3/27/2018
Apples	100	75	3/28/2018
Bananas	150	100	3/28/2018
Oranges	300	260	3/28/2018
Pears	120	80	3/28/2018
Apples	250	200	3/29/2018
Bananas	160	80	3/29/2018
Oranges	270	180	3/29/2018
Pears	80	50	3/29/2018

Assume you want to monitor, for the most recent date, the absolute profit as well as the relative profit per fruit type. In addition, you want to use background colors of the tiles to indicate if the relative profit falls below 30% for any fruit type, or if it exceeds 70%.

Creating a KPI chart from scratch based on the data above results in this KPI:



However, you want the primary values to indicate the absolute net profit instead of the actual sales figures. On the **Value (y-axis)**, specify the following [custom expression](#), which calculates the absolute net profit:

```
Sum([Sales]- [Purchase])
```

The result is shown below. For example, the pears net profit for the most recent date is  $80 - 50 = 30$  (see the data table).



The sparkline now reflects the new expression on the **Value (y-axis)**.

The comparative values in the tiles can be used to display the relative profit. On the **Comparative value** axis, specify the following custom expression, which calculates the relative profit:

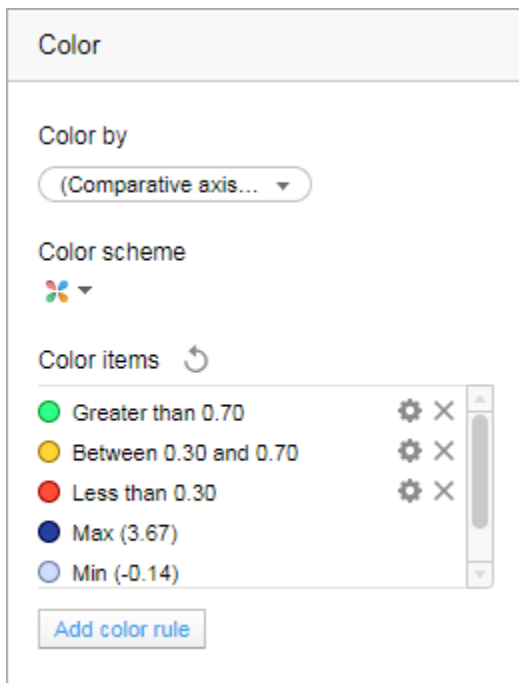
```
(Sum([Sales]) / Sum([Purchase])) - 1
```

The relative profits are added to the chart below. For example, the relative profit for pears is  $80/50 - 1 = 0.60$ .





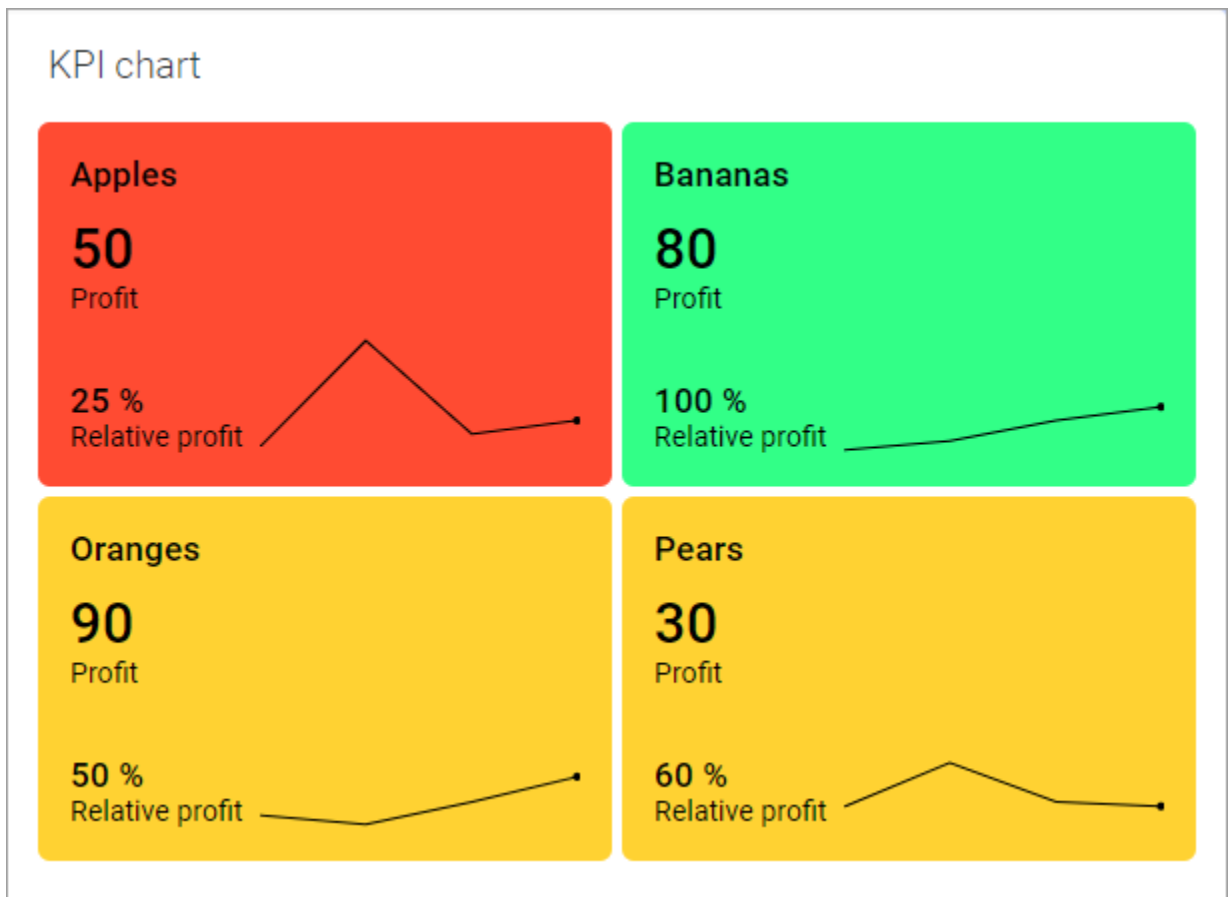
Using different colors of the tile backgrounds, you can easily make it possible to spot fruits whose profit falls below or exceeds certain values. On the **Color** section of the Properties dialog, rules can be added that color the tiles differently.



In this example, you want the background colors to reflect what is calculated on Comparative axis, that is, the relative profit. So **Color by** is set to (**Comparative axis values**), and the tiles below are colored according to the color rules that have been added (see the image above).



You can make the appearance of the KPI cleaner. The expressions used for calculating the primary and comparative values have been [renamed](#) below, and the [formatting has been changed](#).



## Specifying generic KPI chart properties

In a KPI chart, each KPI has its own properties to be specified. However, there are also properties to specify that are generic for the entire KPI chart.

You can specify the visualization title and description, and [minimum tile width](#).

### Procedure

1. Right-click the KPI chart, and select **Properties** in the opened menu.  
The Properties pop-over with a side panel is displayed.
2. In the side panel, select the top item.  
The top item shows the title of the KPI chart.  
The Properties pop-over for the generic settings is displayed.
3. Select the Properties section of interest.

## Changing the layout of the KPI chart

You can customize the layout of the KPI chart. For example, you can choose to start each KPI on a new row, specify the minimum tile width, or how many tiles should at least be visible without having to scroll.

Each of the following steps are optional, however, it is highly recommended to follow the order specified here.

## Procedure

1. Right-click the KPI chart, and select **Properties** in the opened menu.  
The Properties pop-over with a side panel is displayed.
2. In the side panel, select the top item.  
The top item shows the title of the KPI chart.  
The Properties pop-over for the generic settings is displayed.
3. In the pop-over, select **Layout**.  
The **Layout** section is shown.
4. Select the check box **Start each KPI on a new row** to be able to better distinguish the individual KPIs from each other.
  - a) Select **Show titles** to display the title above each KPI
  - b) Adjust the spacing between the KPIs by specifying the number of pixels in **Additional spacing (in pixels)**.
5. Under **Min tile width relative to font size (in pixels)**, specify the minimum tile width that you accept.



If the width falls below this value due to lack of space in a row of tiles when the visualization is resized, tiles will be moved to the next row in the grid. If tiles cannot be moved to the next row or must remain visible due to what has been set in **Min number of tiles per row** and **Min number of visible tiles**, the tiles will be scaled down while keeping the ratio between the tile width and the font size.

The minimum tile width is relative to the font size. If you change the font size, the tile width is automatically increased or decreased by the same multiple as the font size. For example, when you increase the font size from 9 to 18, the tile width doubles automatically. The value specified in **Min tile width relative to font size (in pixels)** however, stays the same. This value only reflects the tile width as the actual number of pixels, when the font size is set to Default. In the installed client, you can change the font size for a specific visualization. In the web client, however, you can only change the Default font size in the [Custom theme settings](#).

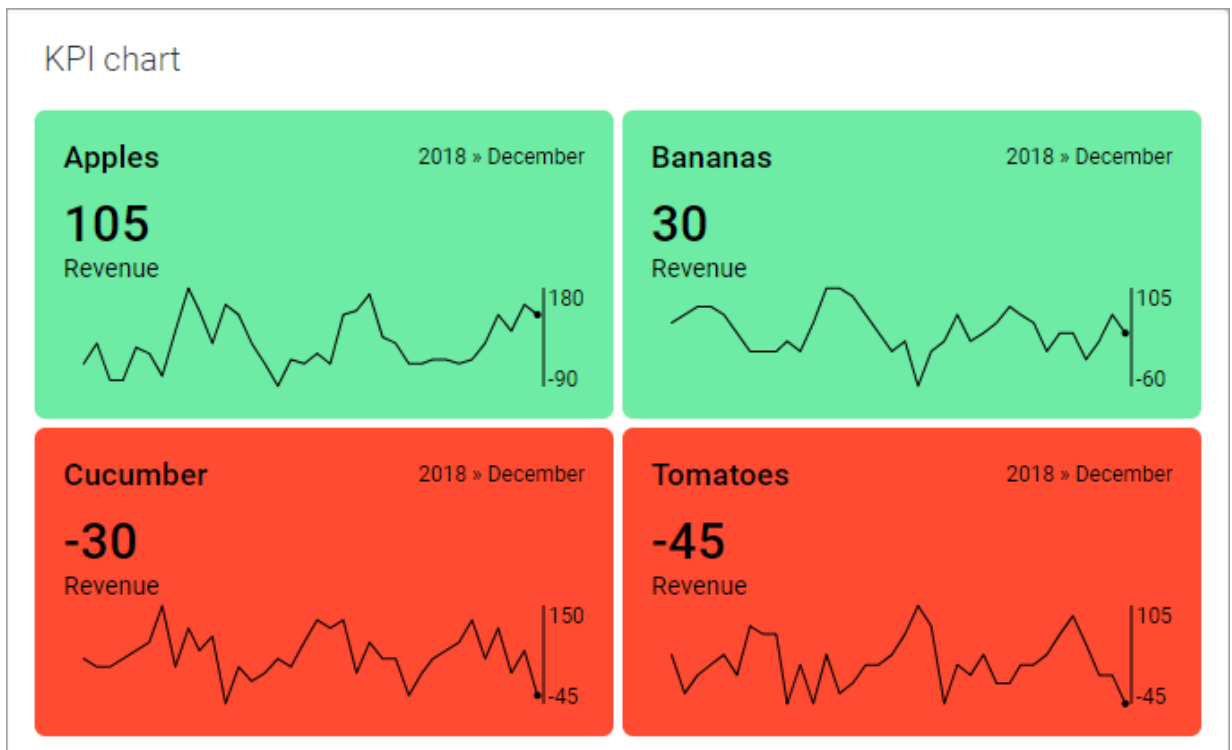
6. Under **Min number of tiles per row**, specify how many tiles must remain in one row when the visualization is resized.
7. Under **Min number of visible tiles**, specify how many tiles must remain visible without having to scroll when the visualization is resized.

## Showing sparklines

Key Performance Indicators, KPIs, are used to monitor performance of a company. They reflect the performance for the most recent time period. If you want to see also the trend over time for a KPI value, you can show a simple line graph, a sparkline, in the KPI tiles.



**Important** The values presented in a sparkline are always the [primary values](#).



The dot ending a sparkline indicates the value of the most recent measure, and when on the time line it was measured. The sparklines can be supplemented with a scale showing the vertical range.

### Procedure

1. Right-click the KPI in the KPI chart, and select **Properties** in the opened menu.  
The Properties pop-over for the selected KPI is displayed. In the side panel, the current KPI is highlighted.
2. In the pop-over, select **Sparkline**.  
The Sparkline section is shown.
3. Make sure the **Show sparkline** check box is selected.
4. To show the vertical range, select the **Show scale** check box.
5. If you want the value 0 to be within the range, select the **Include origin** check box.
6. Click to select whether to use **One scale for all sparklines** in the KPI, or **Multiple scales**.  
Use **One scale for all sparklines** if you want to compare the sparklines under the same conditions for all the tiles within the KPI. The **Multiple scales** option is useful, for example, if the data to present in the different sparklines is of very different magnitudes, and you are, in the first place, interested in seeing the trend for each tile category.

To hide the sparklines, simply clear the **Show sparkline** check box.

### Example

In the image above, multiple scales are used for the sparklines, and origin is not included in the scale.

## Sorting tiles alphabetically

You can sort the categories represented by different tiles in a common alphabetical order. The sorting includes the tiles from all existing KPIs in the KPI chart.

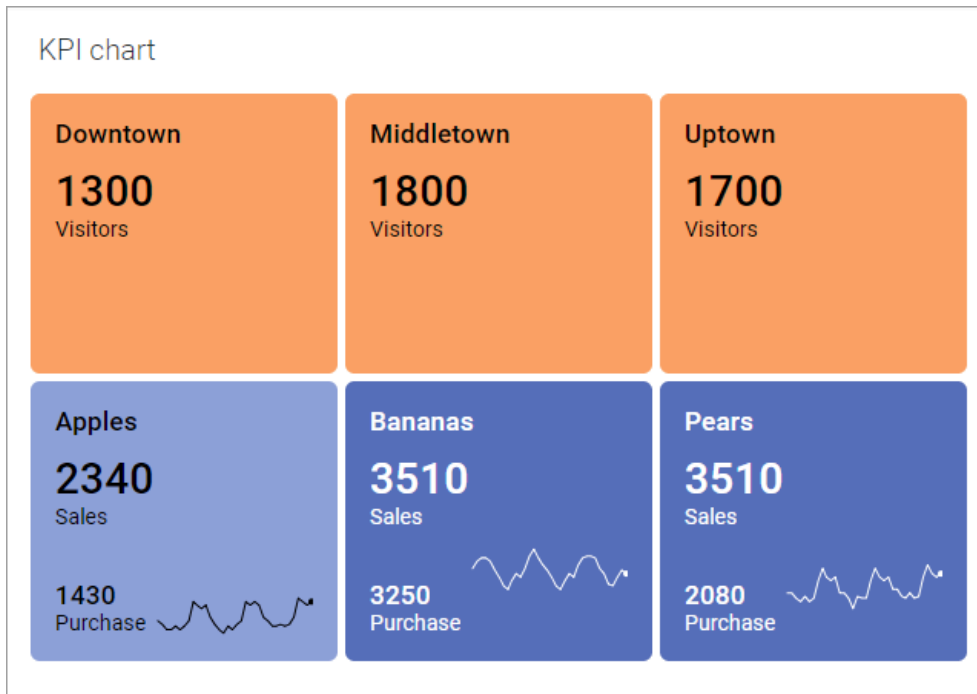
### Procedure

1. Right-click the KPI chart to display the pop-up menu.
2. Select **Sort order > Alphabetical**.

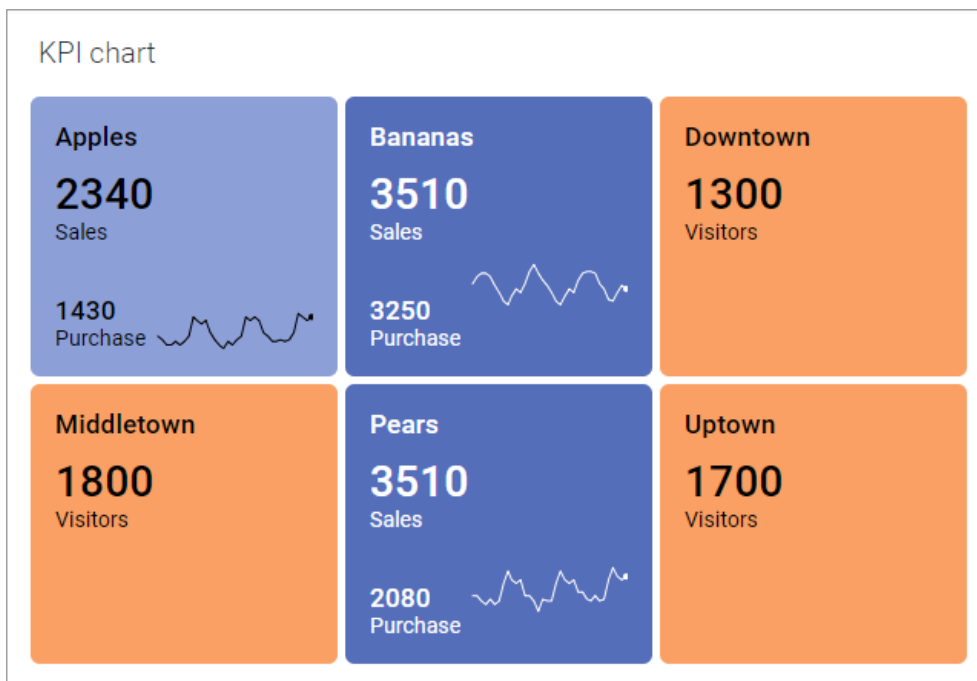
If you want to display the tiles the other way around, select **Sort order > Reversed alphabetical**.

### Example

The KPI chart below contains two KPIs. The first KPI shows the number of visitors at different stores, and the second KPI shows sales and purchase figures for different fruit types. The tiles are not sorted.



Sorting the tiles in alphabetical order results in the KPI chart below. The categories are ordered alphabetically.




## Sorting tiles by performance

You can sort the tiles in a KPI chart in 'Best first' or 'Worst first' order by measures that you specify per KPI. The sorting includes the tiles from all existing KPIs in the KPI chart.

This makes it possible to, for example, gather all tiles that show good performance at the very top of the chart by specifying a 'Best first' order.

### Procedure

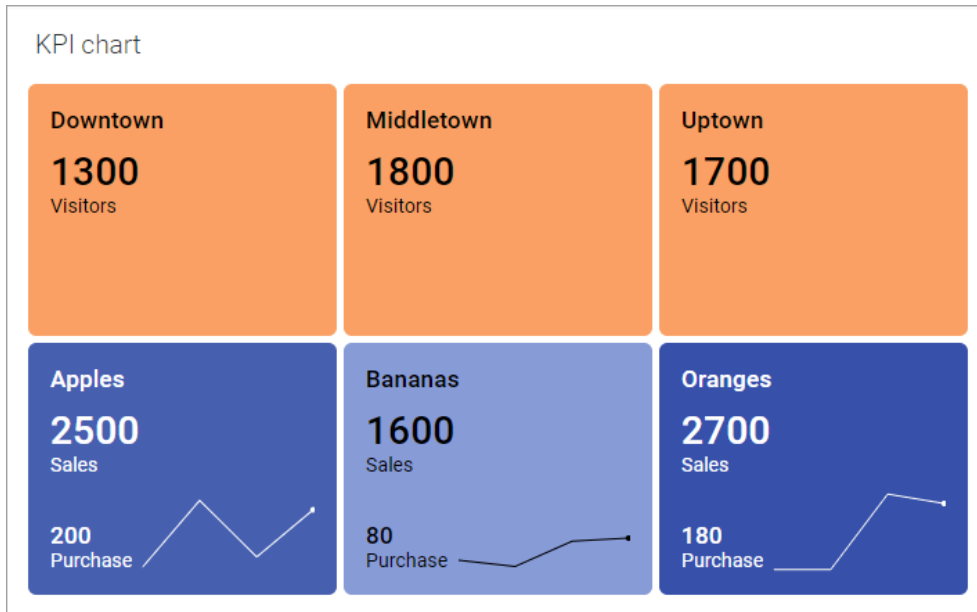
1. Right-click the KPI you want to include in the sorting, and select **Properties** in the opened menu. The Properties pop-over is displayed.
  2. In the pop-over, select **Sorting**. The **Sorting** section is shown.
  3. Under **Sort tiles by**, specify which measure to use for sorting the tiles. You can
    - [select a column](#) and an [aggregation](#) in the column selector.
    - right-click the column selector and click **Custom expression**. This opens a dialog where you can type your own expression.
-  The sorting is only put in effect if the **Best first** or **Worst first** option is used. See the following steps.
4. Repeat step 1 to 3 above until you have specified a sort order for each KPI you want to include in a **Best first** or **Worst first** sorting.
  5. Right-click the KPI chart to display the pop-up menu.
  6. Select **Sort order > Best first** or **Sort order > Worst first**.



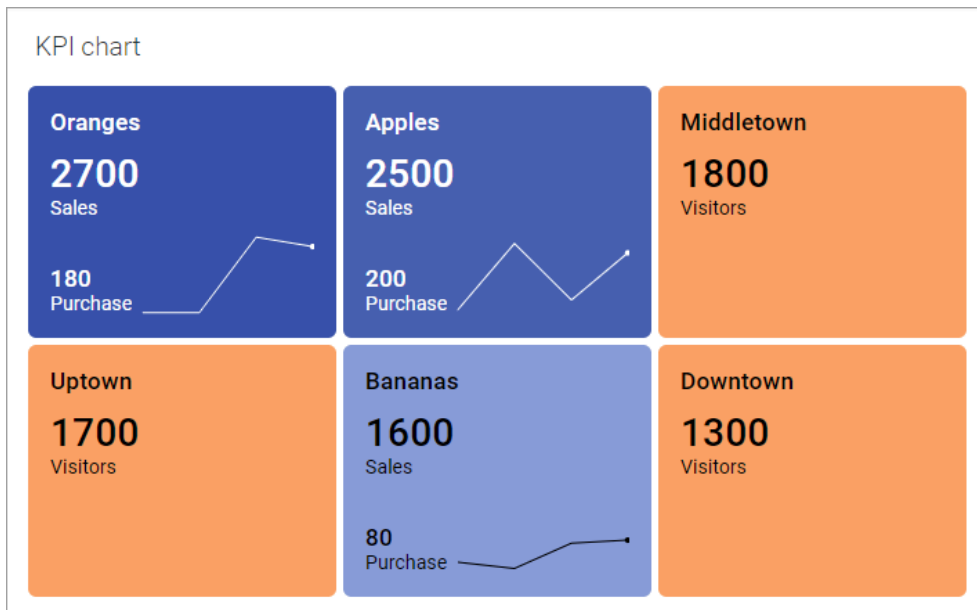
### Example

The KPI chart below contains two KPIs. The first KPI shows the number of visitors at different stores. The **primary values** in the tiles for this KPI are specified to be used for sorting, that is, on **Sort tiles by**, the Visitors column is selected.

The second KPI shows sales figures for different fruit types as primary values in the tiles, and purchase figures as the comparative values. Also in this KPI, the primary values are specified to be used for sorting, that is, on **Sort tiles by**, the Sales column is selected.



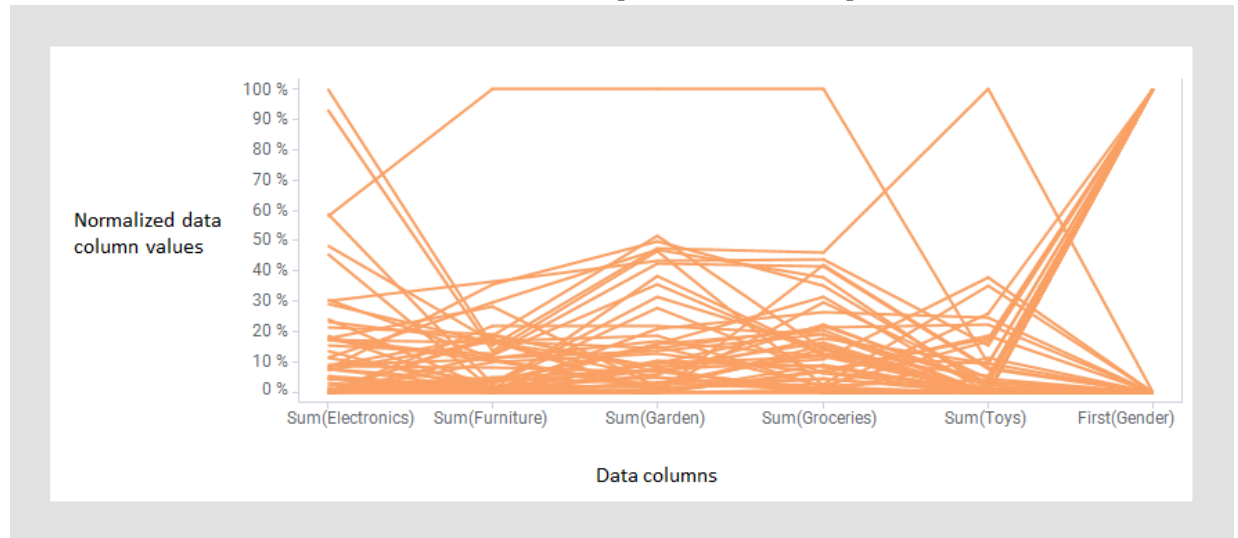
Sorting the tiles using the **Best first** order results in the KPI chart below. The primary values from both the KPIs are sorted starting with the highest value.



## Parallel coordinate plot

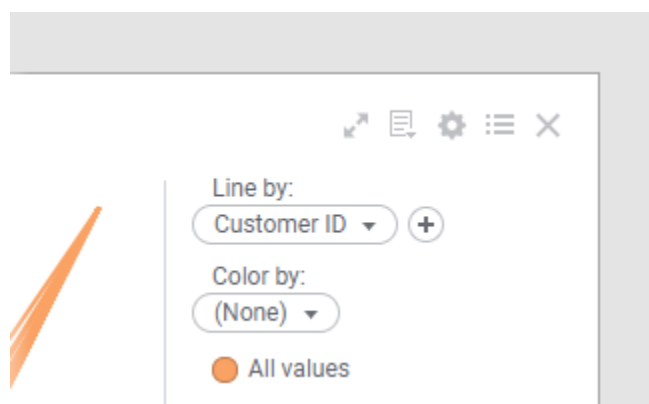
A parallel coordinate plot is used to compare data values which are of completely different types or magnitudes within a single visualization. The values are normalized and then presented as points on a line, or a profile, with one point per data column. This makes parallel coordinate plots similar in appearance to line charts, but the way data is translated into a plot is very different. The visualization is useful also for examining patterns.

The normalized values are expressed as a percentage. The lowest value in a data column is always set to 0%, and the highest value is set to 100%. Values in between are recalculated accordingly. The normalization makes it possible to visualize columns containing values of completely different magnitudes. For example, a column with values between 0 and 1 and a column with values between 0 and 10000 can be visualized at the same time in the parallel coordinate plot.

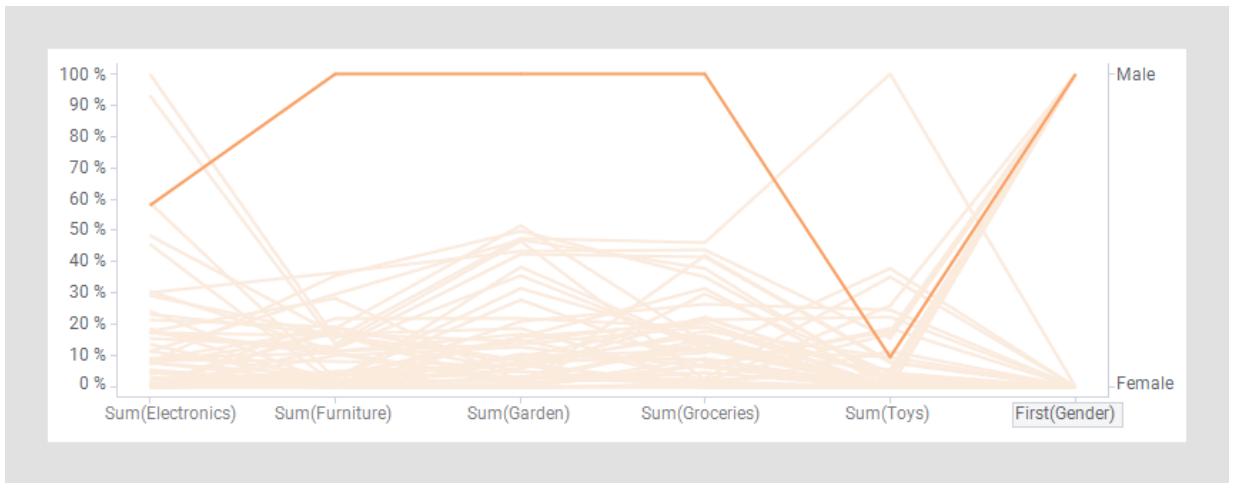


The values can represent aggregated data or non-aggregated data for the particular data point. An aggregated value could be, for example, a sum, an average or the first value in a data column.

Important axes are the **Line by** axis and the **Color by** axis. These axes are used to select the columns whose values you want to represent as lines and you can modify them from the legend.



For example, the parallel coordinate plot above is based on customers' total purchase amounts at different store departments. Each customer is represented by a line, so buying patterns can be examined. To illustrate, see the line below. This customer seems to be a high spender in the first place at the furniture, garden and groceries departments, but also at the electronics department. The toys purchases are relatively low though.



The scale of the various columns is totally separate, so do not compare the height of the line in one column to the height of the line in another column. For example, the actual amount the customer spent at the toys department above can be higher than the amount spent at any other department.

Categorical column values are also possible to visualize. Note the gender column furthest to the right where male and female customers are split into different values on the percentage scale.


## Creating a parallel coordinate plot

The parallel coordinate plot is used to compare data values of different types or magnitudes. All values are presented as points on a line, or profile, with one point per data column. Items that are similar to each other often show a similar profile.

### Prerequisites

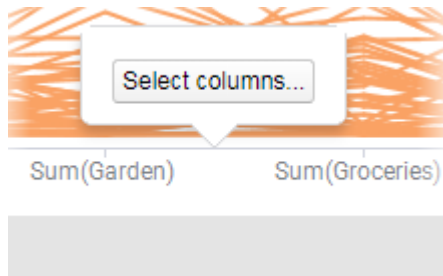
See [Parallel coordinate plot](#) on page 454 for a description of the concept.

### Procedure

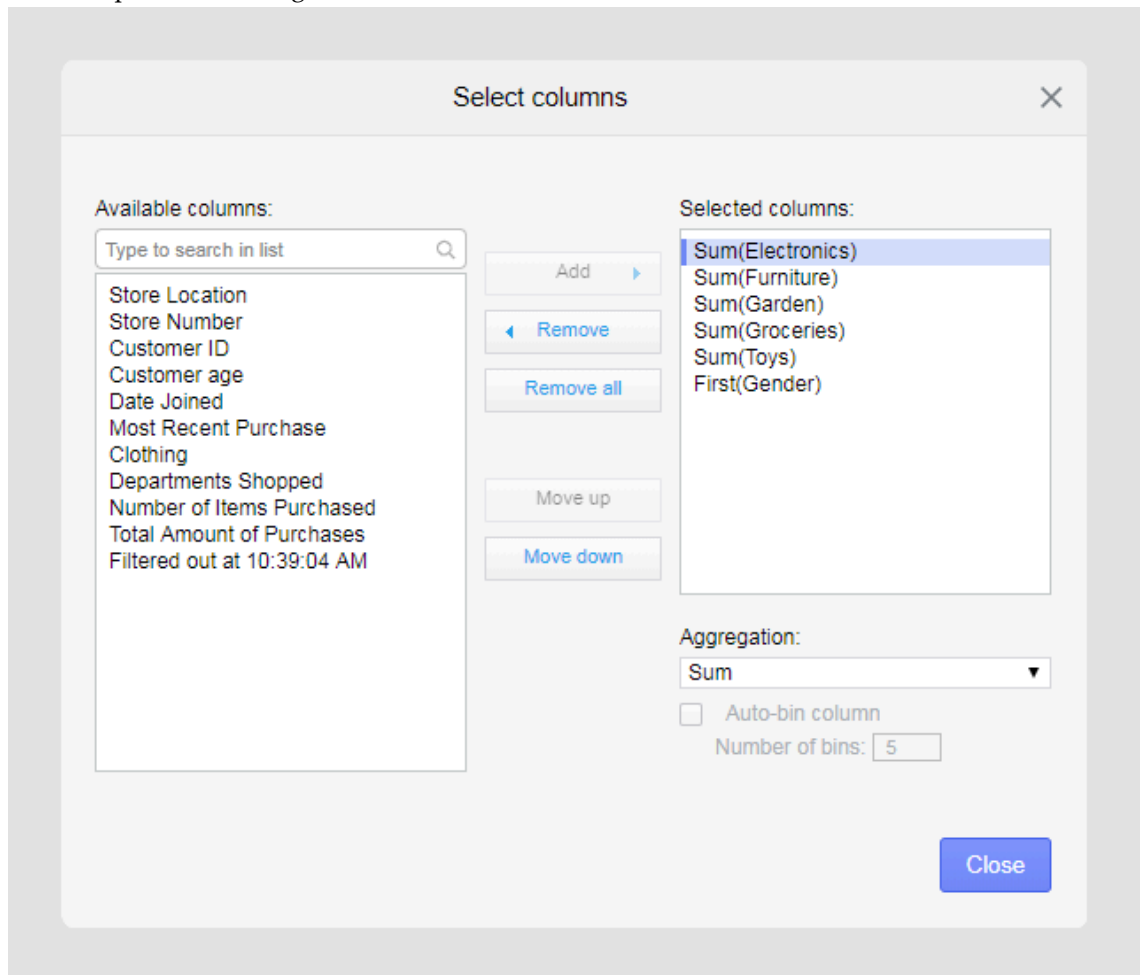
1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Drag the **Parallel coordinate plot** visualization type to the wanted position on the analysis page. A suggestion of a parallel coordinate plot is presented.

3. Select the data columns you wish to include in the visualization.

- a) Click the horizontal axis.  
The **Select columns** button is shown.



- b) Click it to open the Select columns dialog.  
An example of the dialog is shown below.



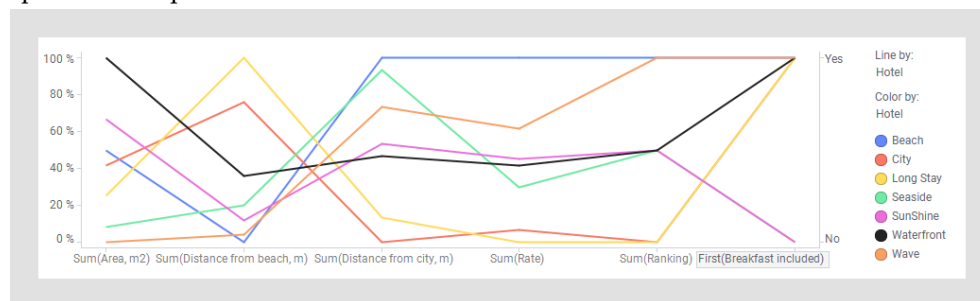
- c) In the dialog, specify the columns of interest, and in which order they should be shown by using the buttons in the middle.
4. Select which aggregation to apply for each of the columns to display.
- a) Select the column in the **Selected columns** list.
- b) In the **Aggregation** drop-down list, select the wanted aggregation type.
- c) Click **Close**.
5. On the **Line** or **Color** axis, select the columns whose values you want to represent as lines.

## Example

In the data table below, details about hotels at a tourist resort are listed. The values in the columns are of completely different types and incomparable to each other.

Hotel	Area, m2	Distance from beach, m	Distance from city, m	Rate	Ranking	Breakfast included
Long Stay	20	1500	400	149	2	Yes
SunShine	25	400	1000	190	3	No
Wave	17	300	1300	205	4	Yes
Beach	23	250	1700	240	4	Yes
Waterfront	29	700	900	187	3	Yes
Seaside	18	500	1600	176	3	No
City	22	1200	200	155	2	Yes

By creating the parallel coordinate plot below with one line per hotel, it is possible to compare the hotels and find a hotel that suits you. Here, the **Color** axis is used to split the data per hotel.



Perhaps Waterfront is a good choice, because you get a large room for a fair rate, and it is located not too far away from neither the beach nor the city. In addition, breakfast is included in the price, and the ranking is OK.



The actual values in a column can be displayed along the right-hand side of the visualization. Click the X-axis label for the column you are interested in.

## Displaying actual values in the parallel coordinate plot

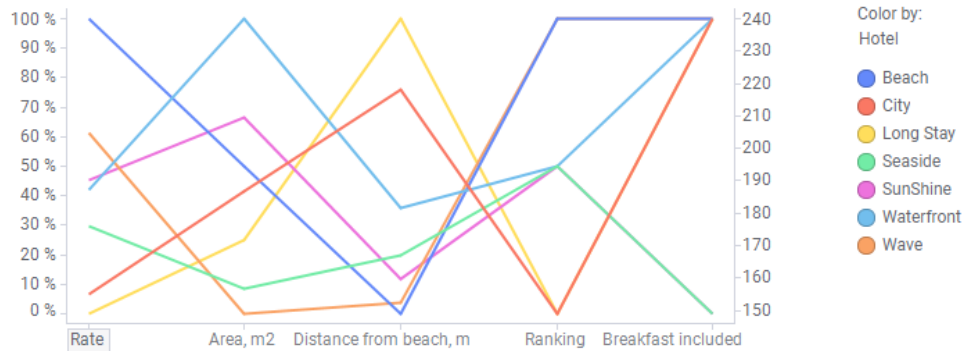
In the parallel coordinate plot, the values are normalized and expressed as a percentage. The percentage scale is displayed to the left in the visualization. It is possible, though, to display also a scale for actual values. The scale can be displayed to the right in the visualization for one column at a time.

### Procedure

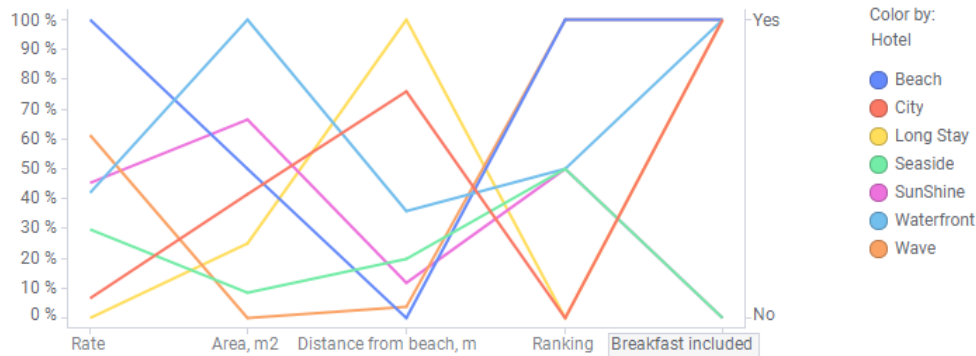
- On the horizontal axis in the parallel coordinate plot, click the label for the column whose actual scale you want to display.

## Example

In the parallel coordinate plot below, the 'Rate' label has been clicked. The actual scale for the rates is displayed to the right of the lines.



Clicking the 'Breakfast included' label instead shows the actual values in the column.



## Summary table

The summary table is a visualization that summarizes statistical information about data in table form. The information is based on one data table in the analysis. When configuring the summary table you can choose which measures to show (such as mean, median, and so on), as well as the columns on which to base these measures.



Summary tables must be authored in the installed client.

Read more about different measures that can be shown in [Aggregations and statistical measures](#) on page 920.

As with all visualizations, the default configuration for the summary table is to automatically update the values displayed to reflect the current selection.

Column	Sum	Avg	Min	Max	Median	StdDev
Sales	3956.00	18.31	8	35	17.50	6.72
Cost	3194.00	14.79	6	29	13.00	6.35

All visualizations can be configured to show data limited by one or more markings in other visualizations only (details visualizations). Summary tables can also be limited by one or more filterings. Another alternative is to configure a summary table without any filtering at all.

You can rearrange or sort the different columns in the summary table in the same way as with other table visualizations.

## Box plot

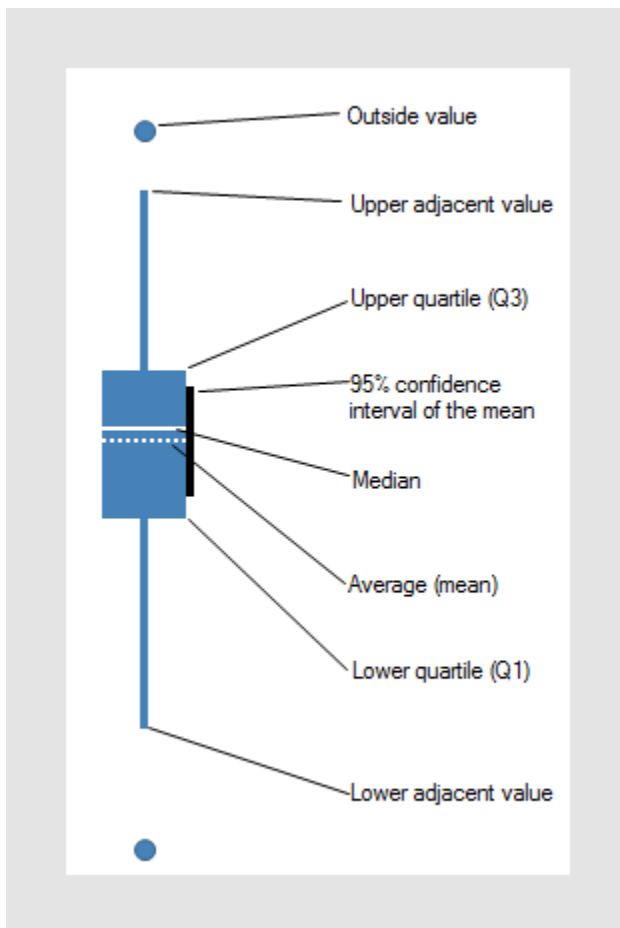
Box plots are graphical tools to visualize key statistical measures, such as median, mean and quartiles.



Box plots must be authored in the installed client.

A single box plot can be used to represent all the data. It is also possible to show separate statistics for subsets of the data by selecting a column for the X-axis.

The individual box plot is a visual aid to examining key statistical properties of a variable. The diagram below shows how the shape of a box plot encodes these properties. The range of the vertical scale is from the minimum to the maximum value of the selected column, or, to the highest or lowest of the displayed reference points.

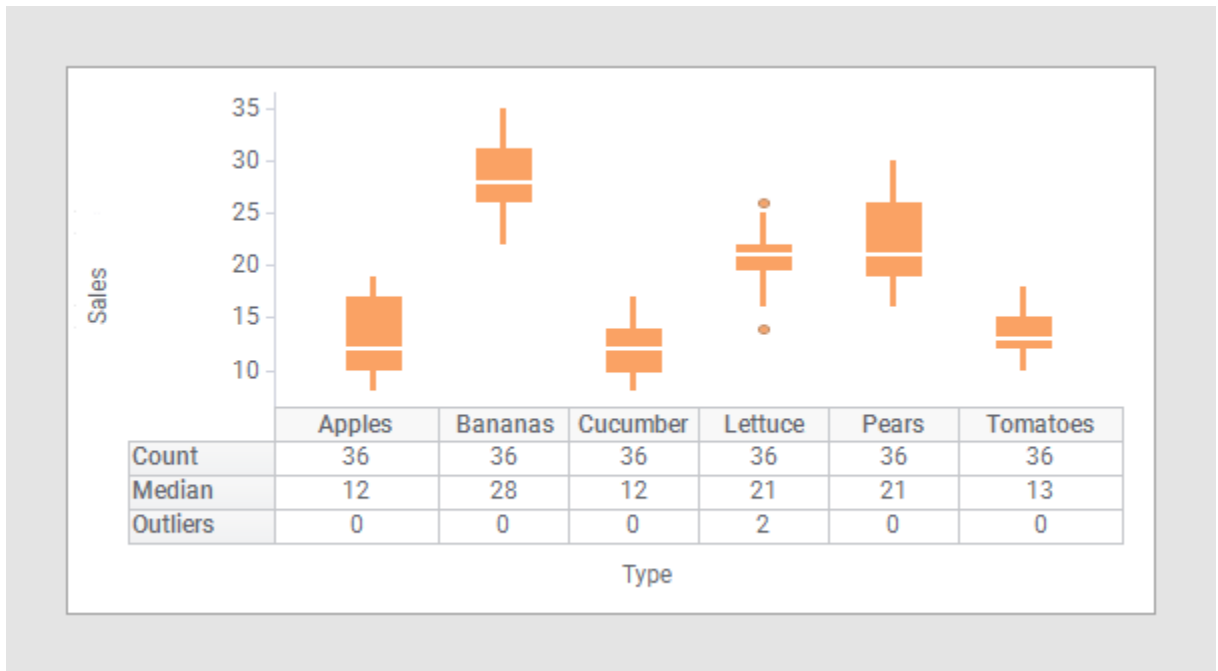


Which reference points should be shown in the box plot is specified upon configuration of the visualization. A reference point can be indicated by either a marker or a line, and you can specify its color and shape. For details of each measure, see [Aggregations and statistical measures](#) on page 920.

The axis selectors control which column is mapped to which axis. See [Selecting columns on an axis](#) on page 479 for more information about how they work.

The Y-axis should be set to the column or columns on which the statistical measures should be based.

The X-axis can be set to any column. However, because a separate plot will be drawn for each unique value, the column or hierarchy should not contain too many unique values. To summarize the data in a single plot, select (None) on the X-axis. If multiple columns are used on the Y-axis, '(Column Names)' must be used either on the X-axis or in one of the trellis options.



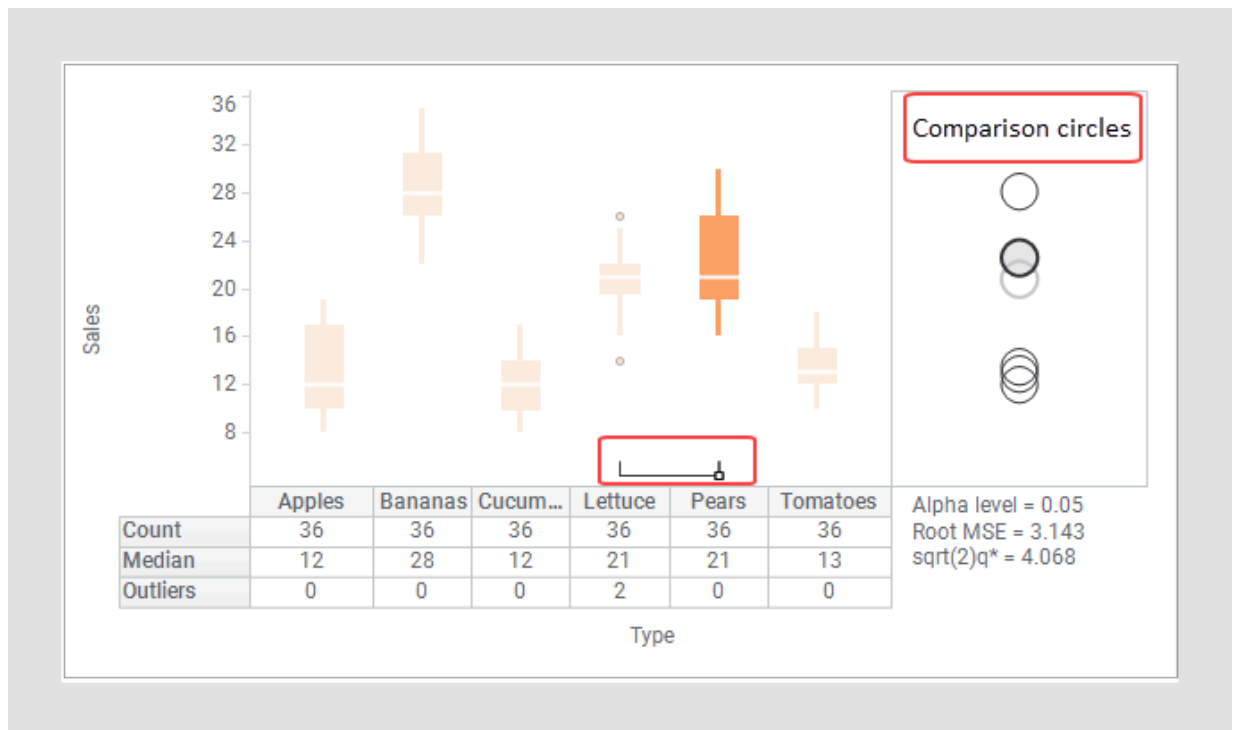
All visualizations can be configured to show data limited by one or more markings in other visualizations only (details visualizations). Box plots can also be limited by one or more filterings. Another alternative is to set up a box plot without any filtering at all.

## Comparison circles

The drawing of comparison circles is a way to show whether the mean values for various categories (boxes in the box plot) are significantly different from each other. The circles are drawn with their centers at the mean value for the box to which it corresponds.

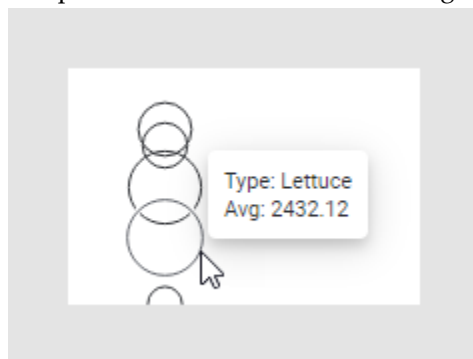
If the circles for different groups do not overlap, the means of the two groups are generally significantly different. If the circles have a large overlap, the means are not significantly different.





In the example above, the sum of sales is shown for a number of different fruits and vegetables. The box for Pears has been marked, which is also indicated in the corresponding comparison circle. The marked comparison circle is shown with a darker border and a transparent fill. By looking at the comparison circles or the little relation indicator at the bottom of the visualization area (circled in red on the image), one can see that the sum of sales for Lettuce does not display a significantly different group mean, whereas the group means for all other fruits and vegetables are significantly different from that of Pears. The square in the relation indicator indicates the marked box and the lines in the relation indicator extend to any boxes that are not significantly different from the marked one.

You can also highlight a comparison circle by highlighting its box plot or vice versa. The highlighted comparison circle is drawn with a slightly lighter color:



You can resize the area containing the comparison circles by placing the mouse pointer over the vertical line separating the circles from the box plots and dragging the handle to the desired position.

### Comparison circles algorithm

The drawing of comparison circles is a way to show whether or not the group means for all pairs are significantly different from each other. The Tukey-Kramer method is used for the calculation. Each group (each box plot) gets a circle where the center of the circle is aligned with the group mean value.

The radius of the circle,  $r_i$ , is calculated as follows:

$$r_i = \left| \frac{q}{\sqrt{2}} \right| \sqrt{\frac{MSE}{n_i}}$$

where

- MSE is the mean standard error (the pooled sample variance) for each box:

$$MSE = \frac{1}{v} \sum_{j=1}^K (std)^2 (n_j - 1)$$

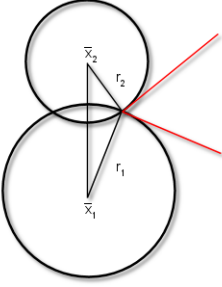
- v is the degrees of freedom:

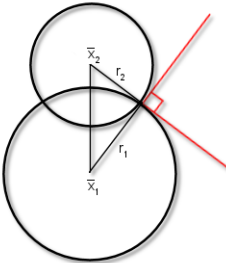
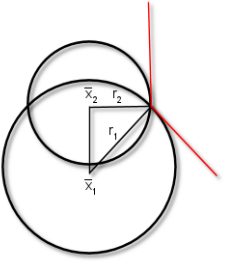
$$v = \sum_{j=1}^K (n_j - 1)$$

- $n_j$  is the number of records in the group (count)
- K is the number of groups
- $q = \sqrt{2} |q^*|_{\alpha, K, v}$  where  $q^*$  is the critical value and q is the upper alpha quantile of the Studentized range distribution with K groups and v degrees of freedom. The alpha level is specified in the Visualization Properties dialog. For details on how the quantile is calculated, see [Aggregations and statistical measures](#) on page 920. The value of q is calculated and updated each time the filtering of the axis is changed.

If the circles for different groups do not overlap (or that the external angle of intersection is less than 90 degrees) the means of the two groups are generally significantly different. If the circles have a large overlap, the means are not significantly different.

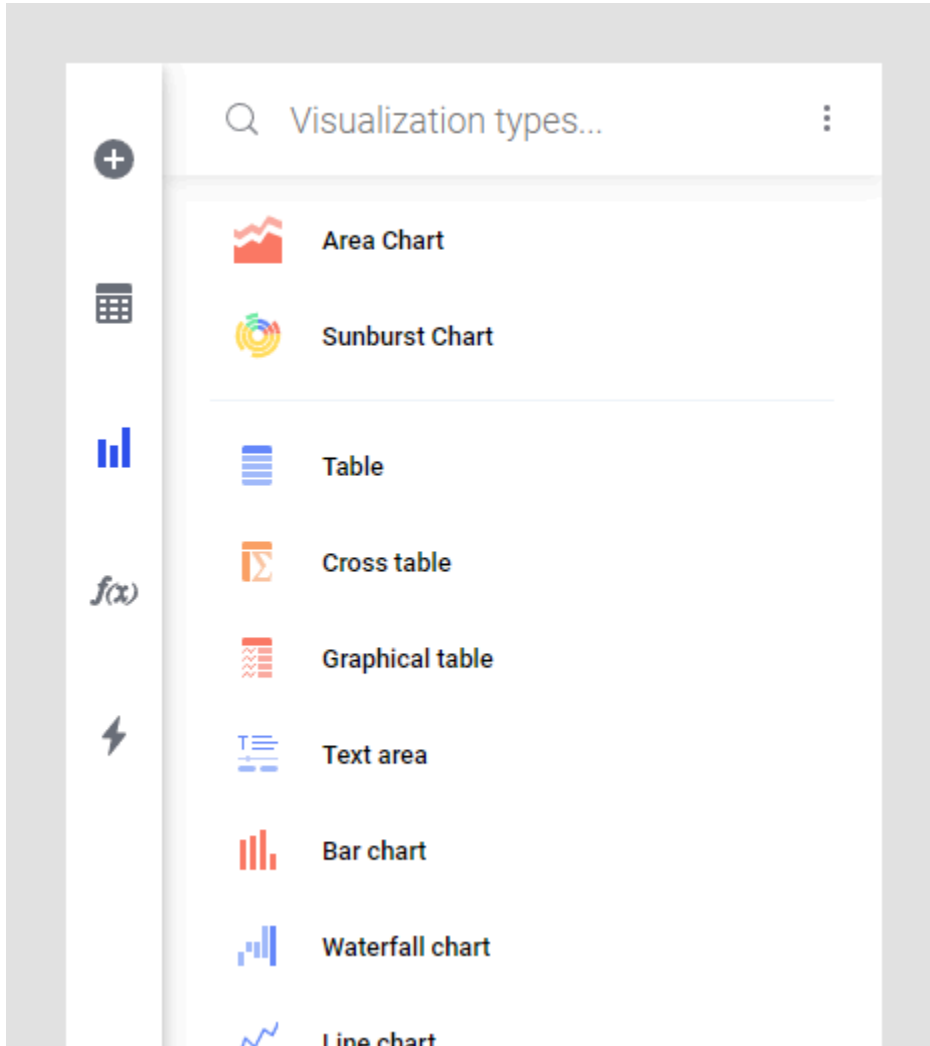
The explanation to why the overlap defines whether or not group means are significant can be deduced with the Pythagorean Theorem.


Comparison circles	Mathematical expression	Interpretation
	$ \bar{x}_1 - \bar{x}_2  > \sqrt{r_1^2 + r_2^2}$	The groups are significantly different.

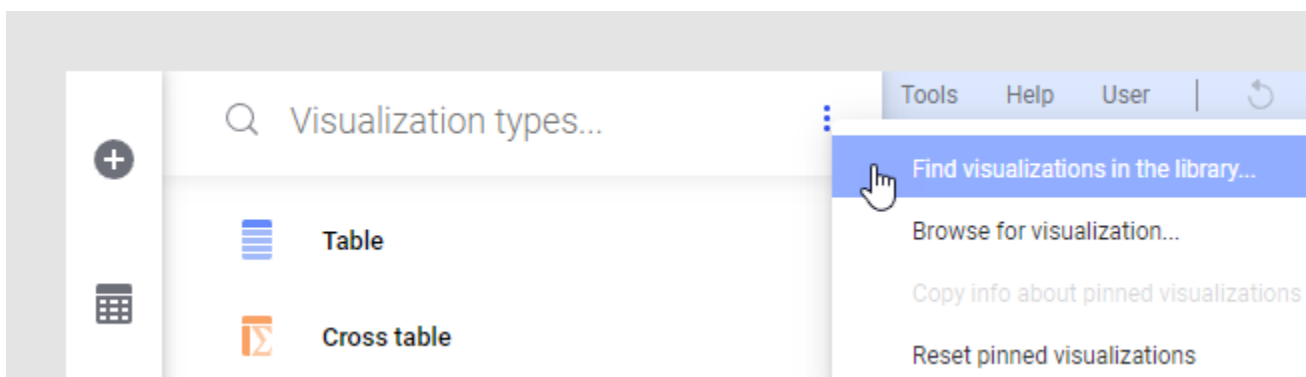
Comparison circles	Mathematical expression	Interpretation
	$ \bar{x}_1 - \bar{x}_2  = \sqrt{r_1^2 + r_2^2}$	Borderline significantly different.
	$ \bar{x}_1 - \bar{x}_2  < \sqrt{r_1^2 + r_2^2}$	The groups are not significantly different.

## The Visualization types flyout

The Visualization types flyout gives you an overview of all the visualization types that are available to you. You can add a visualization to your analysis by clicking the visualization or using drag-and-drop.



You can search for visualizations in the library, by opening the menu  at the top of the flyout and selecting **Find visualizations in the library**.



A search is performed in the library to show all available visualization mods that you can pin to the flyout (type:visualization). You can pin your favorite visualization mods to the flyout to have easy access to them later on. An administrator can pin visualization mods to the flyout using a preference

setting. See [Pinning visualization mods to the visualization flyout](#) on page 465 for more information about pinning.

See also [Saving a visualization mod to the library](#) on page 466 and [Opening a visualization mod that is saved locally](#) on page 467.



## Pinning visualization mods to the visualization flyout

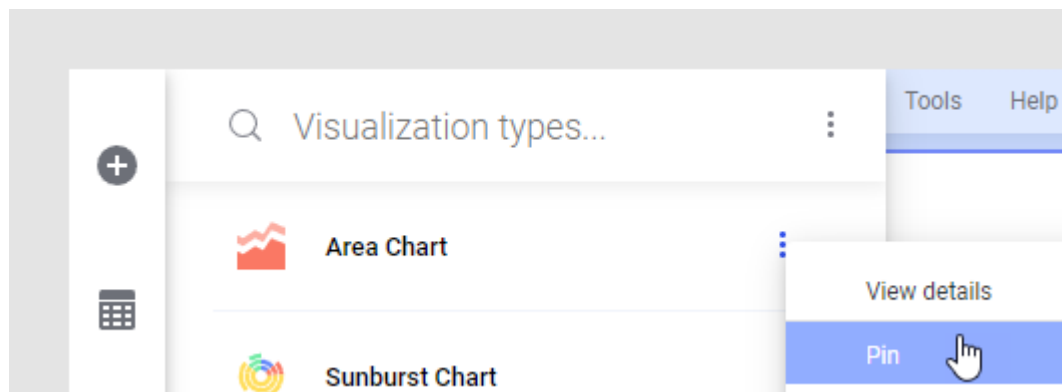
Visualization mods that have been saved to the library can be pinned, which means they are added next to the list of native visualizations in the Visualization types flyout. Once a visualization mod is part of the visualizations list, it becomes available for selection also in future analyses. In addition, the visualization mod will appear as a choice whenever you select a visualization type, for example, when you want to switch to another visualization or create a details visualization.


### Prerequisites

The visualization mod is saved in the library. See [Saving a visualization mod in the library](#).

### Procedure


1. On the authoring bar, click  to open the Visualization types flyout.  
The upper part of the flyout, above the line, lists visualization mods that are not yet added to the list of pinned visualizations.
2. Hover over the visualization mod, click options , and select **Pin**.



If the visualization mod is not available in the flyout, you can click options  in the Visualization types search field at the top of the flyout, and select **Find visualizations in the library**. Locate and right-click the visualization mod of interest, and select **Pin to visualizations**.

The visualization mod is shown below the line, in the lower part of the flyout.



To remove the visualization mod from the Visualization types flyout, locate the visualization to unpin, click the  button, and select **Unpin**.

## Pinning visualization mods to the flyout using administrator preferences

As an administrator with access to the Administration Manager, you can pin visualization mods to make them available by default for certain user groups.

### Prerequisites

You are an administrator with access to the installed Spotfire client.

## Procedure

1. As an administrator, follow the steps above to pin visualization mods to the flyout, so that the selection of visualizations is as you want it to be for the group.
2. When you are satisfied, in the Visualization types flyout, click the menu at the top right of the flyout and select **Copy info about pinned visualizations to clipboard**.  
This action copies the current layout of the flyout to the clipboard in a format that is directly suited to be pasted into the preference as described in the next steps.
3. In the installed client, click **Tools > Administration manager** and go to the **Preferences** tab.
4. Select the group for which to change the preferences and, in the lower right part of the Administration Manager, click **Edit**.
5. Expand **Application**, click **Mods** and, next to **Pinned visualization mods**, paste the previously copied string.  
See the [Administration Manager](#) help for more information about preferences.
6. Click **OK** and **Close** the Administration Manager.

## Saving a visualization mod to the library

You can save a visualization mod to the library. Once saved to the library, the visualization mod can be added to analyses and also pinned to the visualization flyout.





You can search the library to find visualization mods.

### Prerequisites

You must have the license feature Save Visualization Mod to Library, the analysis must be in **Editing** mode, and the visualization mod you want to save must be available in the analysis but not in development.

### Procedure

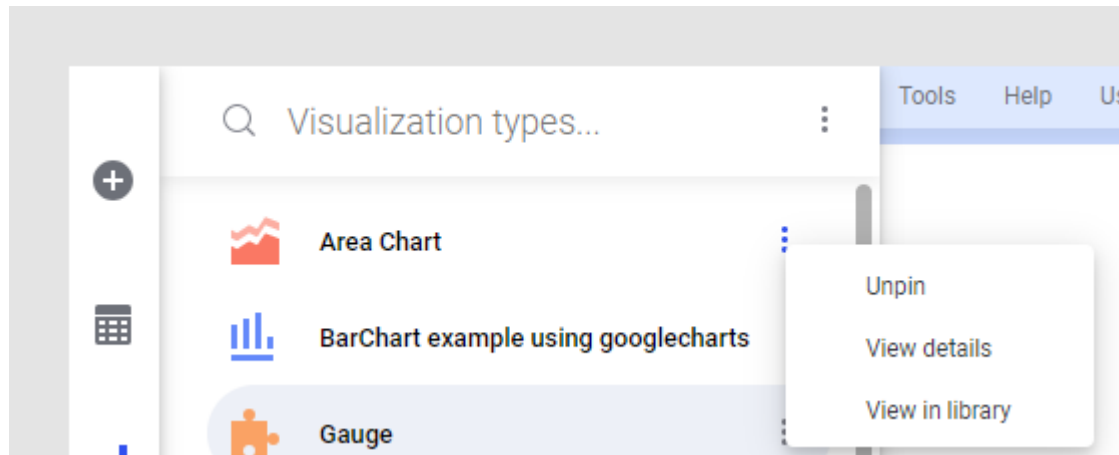
1. On the authoring bar, click **Visualization types**  to open the flyout.
2. Locate the visualization mod you want to save, and click the menu to the right .
3. Select **Save as library item**.  
The **Files and data** flyout is opened.

4. Select a folder to save the mod in, type a name in the bottom part of the flyout, and click **Save**.



If you have the Develop mods license feature, you can also save the visualization mod to the library from the Mods development tool. Make sure the mod is disconnected, and then select **Save as > Library item**.

To find where in the library a visualization mod is located, open the Visualization types flyout, click the menu to the right of the visualization mod, and select **View in library**.



## Opening a visualization mod that is saved locally

A visualization mod can be saved as a .mod file and then distributed between users, via, for example, email or by publishing it on websites. You can download and save such a .mod file locally, and then open the visualization mod in an analysis.




When allowing custom code to run within an analysis, it is important to consider security. For more information, see [Manage trust](#).

### Prerequisites

Make sure data is loaded into the analysis.

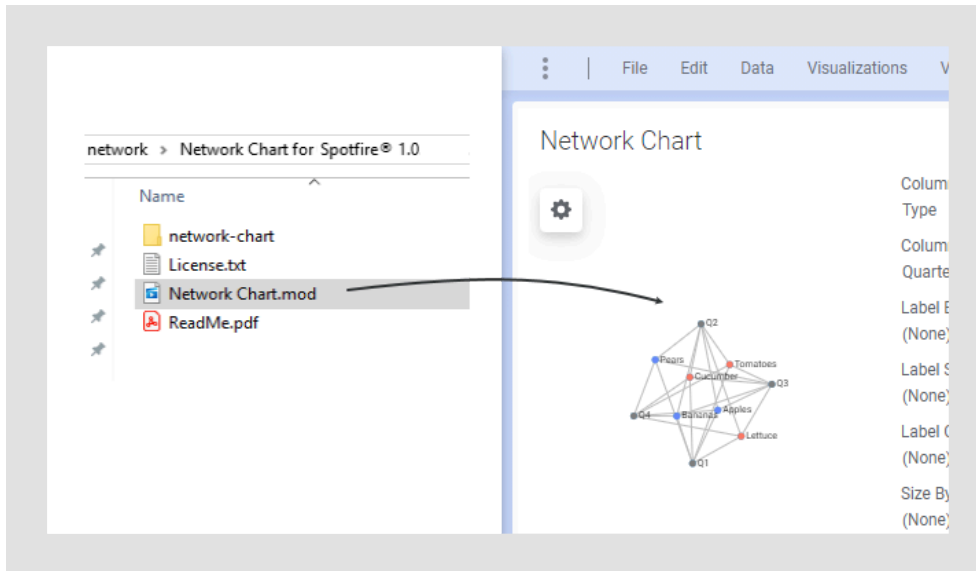
### Procedure

1. On the authoring bar, click **Files and data** . The **Files and data** flyout opens.
2. In the flyout, click **Browse local file**, and locate the .mod file.
3. Right-click the .mod file, and click **Open**. The visualization mod is added to the analysis page.

Alternatively, drag the .mod file to the analysis page as illustrated in the example below.

## Example

The Network Chart.mod file is saved locally. Dragging the file to an analysis results in an opened Network Chart.



## Duplicating a visualization

A copy of a visualization can be created to be used as starting point for creating another similar visualization.

### Prerequisites

The visualization to copy is created.

### Procedure

1. Right-click the visualization.
2. In the pop-up menu, select **Duplicate visualization**.  
An identical visualization appears on the page.

### What to do next

You can adjust the duplicated visualization to suit your needs.



The duplicated visualization can be placed on another page in your analysis. Click the visualization title, drag it to the title of the page where to place the visualization. Then, on the opened page, move the cursor to the position where to drop the visualization. Possible positions are indicated in gray.

## Creating a trellised visualization

A visualization that is trellised is split into a number of panels, where each panel represents a subset of the data. Using trellised visualizations, you can spot similarities and differences between the subsets of data, or within the subsets.

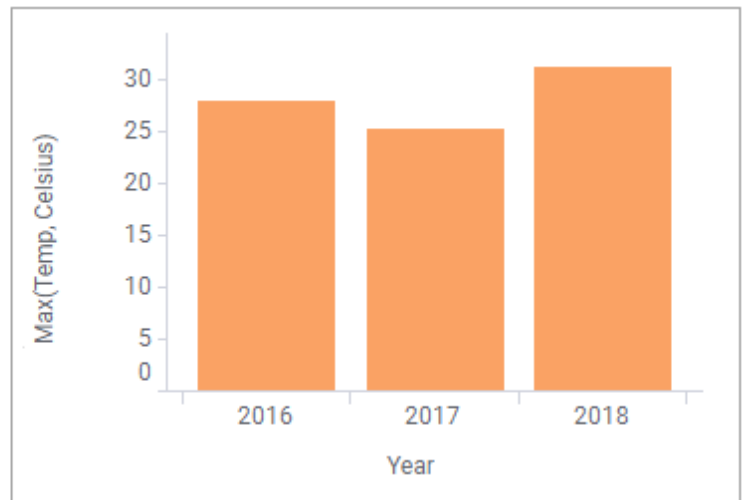
The subsets can be defined by splitting the data into categories that are available in a data column, alternatively by splitting it per data column, and then each category or column is displayed in a panel of its own.



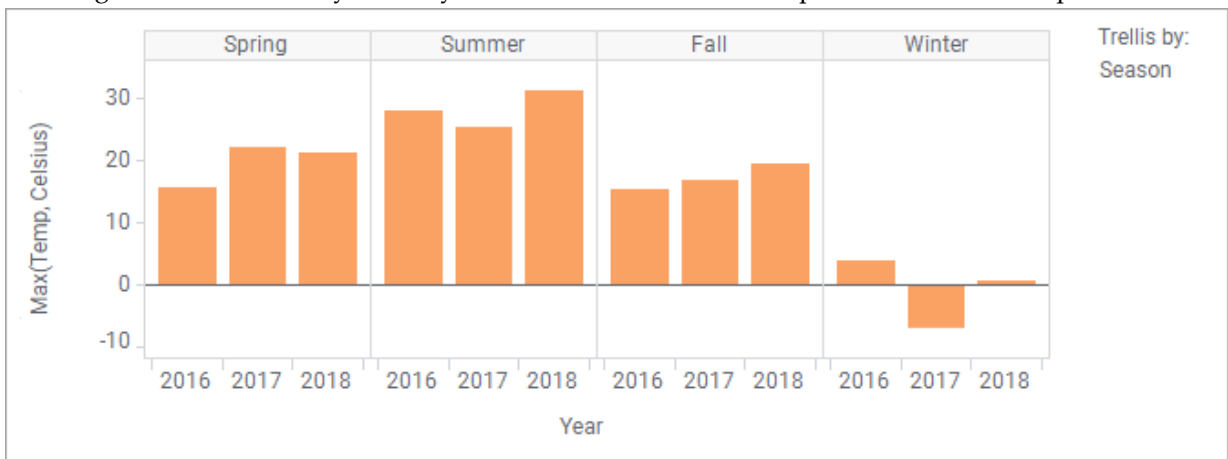
For splitting data per data columns, see [Visualizations trellised by data columns](#).

For example, the image below shows temperature data that is gathered for the different seasons during a three-year period. The maximum temperature per year is displayed in a bar chart.

Year	Season	Temp, Celsius
2016	Winter	3.90
2016	Spring	15.60
2016	Summer	27.80
2016	Fall	15.30
2017	Winter	-6.70
2017	Spring	22.00
2017	Summer	25.20
2017	Fall	16.70
2018	Winter	0.50
2018	Spring	21.00
2018	Summer	31.10
2018	Fall	19.50



Assume you want to compare the highest temperature per season category across the years. By trellising the visualization by season you can view the seasonal temperatures in different panels.



The panels above are placed horizontally, but you can also place them vertically, or evenly distributed in a matrix.



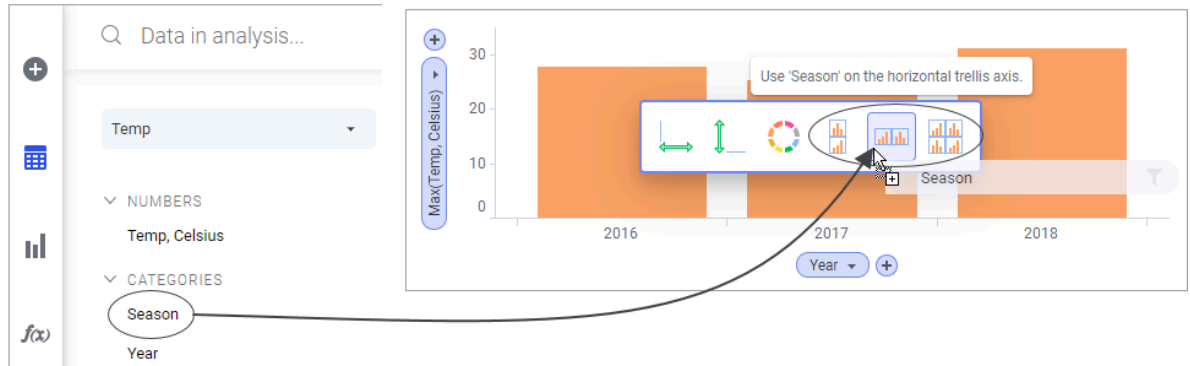
Table visualizations, cross tables, and summary tables cannot be trellised.

### Prerequisites

A visualization that is not in a table format is created.

## Procedure

- In the **Data in analysis** flyout, click the column containing the categories you want to split the visualization by, and drag it on any of the trellis drop targets.



The visualization splits into panels.



In the Properties popover for the visualization, the **Trellis** section, more [trellis options](#) are available.

## Visualizations trellised by data columns

Trellised visualizations are split into panels, where the panels display various subsets of the data table. For example, what is shown in a panel can be based on values from a specific numerical column only.

Subsets of data can be defined by splitting the data into categories that are available in a data column. Creating trellised visualizations with panels that compare categories is described in [Creating a trellised visualization](#).

However, the following examples show trellised visualizations with panels based on certain data columns. In both examples, the data table below is used.

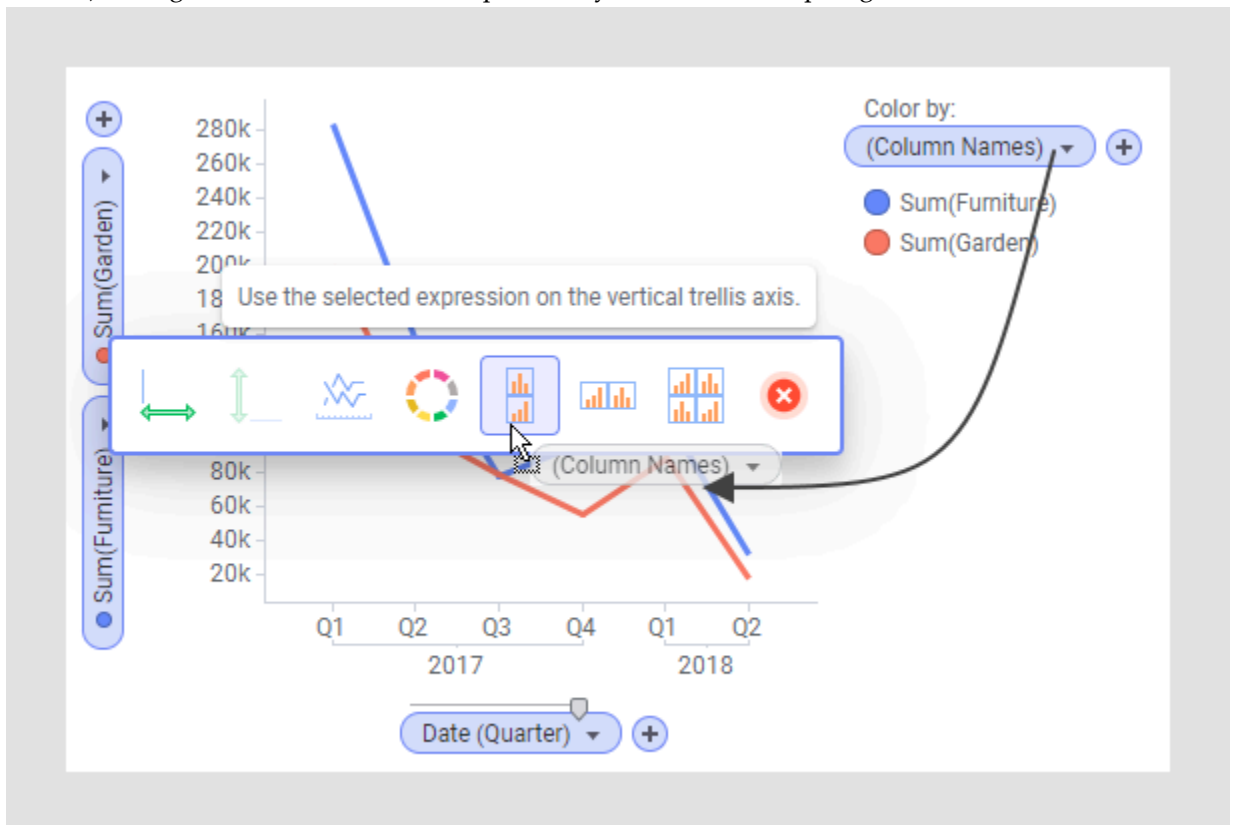
Customer ID	Store location	Customer age	Date	Furniture	Garden
SSMM55001	Los Angeles	61	7/21/2017	3488	6999
SSMM55002	Seattle	43	7/23/2017	5774	7156
SSMM55003	Boston	55	7/23/2017	1369	731
SSMM55004	Boston	41	7/23/2017	1362	1431
SSMM55005	New York	68	7/23/2017	5865	5429
SSMM55006	New York	56	7/23/2017	524	467
SSMM55007	Los Angeles	74	7/23/2017	0	467
SSMM55008	New York	77	7/23/2017	238	0
SSMM55009	Los Angeles	62	7/23/2017	0	0
SSMM55010	Boston	65	7/24/2017	0	747
SSMM55011	Boston	54	7/25/2017	0	405

## Example

The line chart shows sales over time at the furniture and garden departments. How to create this type of line chart is described in [Creating a line chart](#).



You can choose to show the lines for the two data columns in separate panels. Simply drag the (Column Names) setting on the color axis and drop it on any of the trellis drop targets in the visualization.

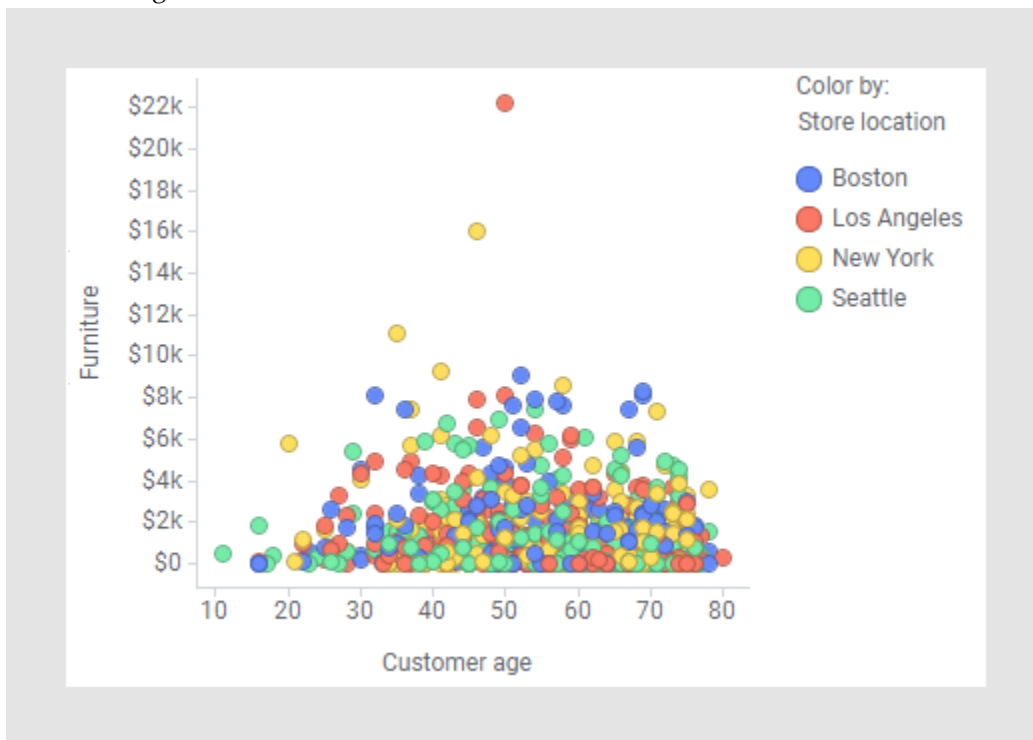


The result, when placing the panels vertically, is shown below.



### Example

The scatter plot shows how the furniture sales at different store locations are spread across the customers' ages.



If another numerical column is added to the Y-axis, and the color axis is already occupied, the two data columns, furniture and garden, will be displayed in separate panels as shown below.



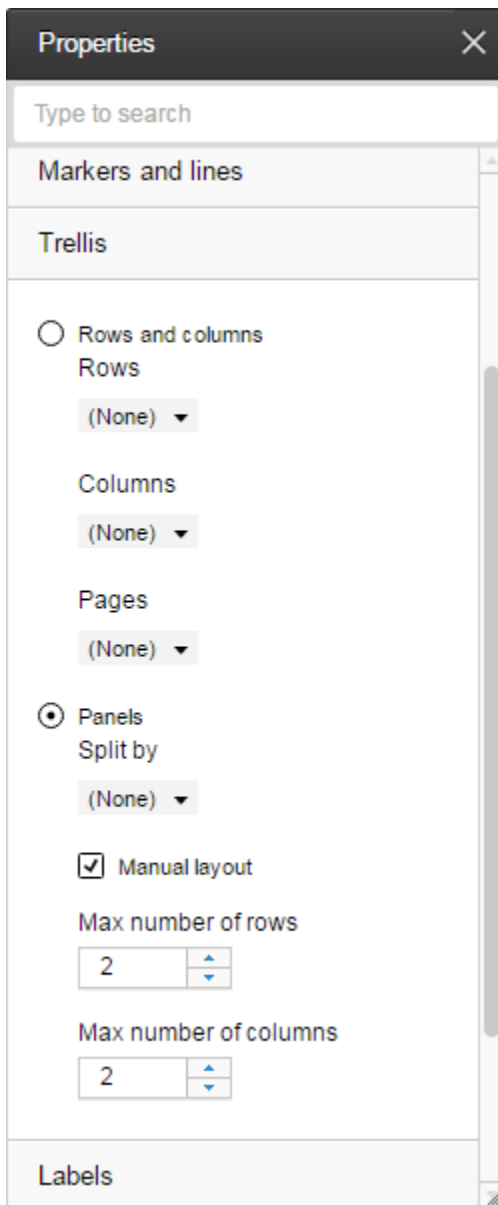
## Trellis options

A visualization that is trellised is split into a number of panels, where each panel represents a subset of the data. Using trellised visualizations, you can spot similarities and differences between these subsets.

When [creating a trellised visualization](#) using the drop targets, you can arrange the trellis panels in either only rows or in only columns, alternatively as a matrix:



These layout options can also be accessed in the **Trellis** section of the Properties popover for the visualization. From there, you have more options to arrange the trellis panels:



**Properties** [X]

Type to search

Markers and lines

Trellis

☐ Rows and columns

Rows  
(None) ▾

Columns  
(None) ▾

Pages  
(None) ▾

☒ Panels

Split by  
(None) ▾

☒ Manual layout

Max number of rows  
2 ▴ ▾

Max number of columns  
2 ▴ ▾

Labels

For example, you can, using the **Rows and columns** option, split the data further by selecting different columns in the **Rows** and **Columns** drop-down lists.

By selecting a column in the **Pages** drop-down list, the different trellis panels in the visualization are treated as "pages" within the visualization. Only one trellis panel is visible at a time. Then you use a scroll bar to view the other panel "pages".

In case you want the layout of a trellis matrix to be different, you can customize it. Via the **Manual layout** check box in the **Panels** option, you specify the wanted maximum number of rows and columns in the matrix.

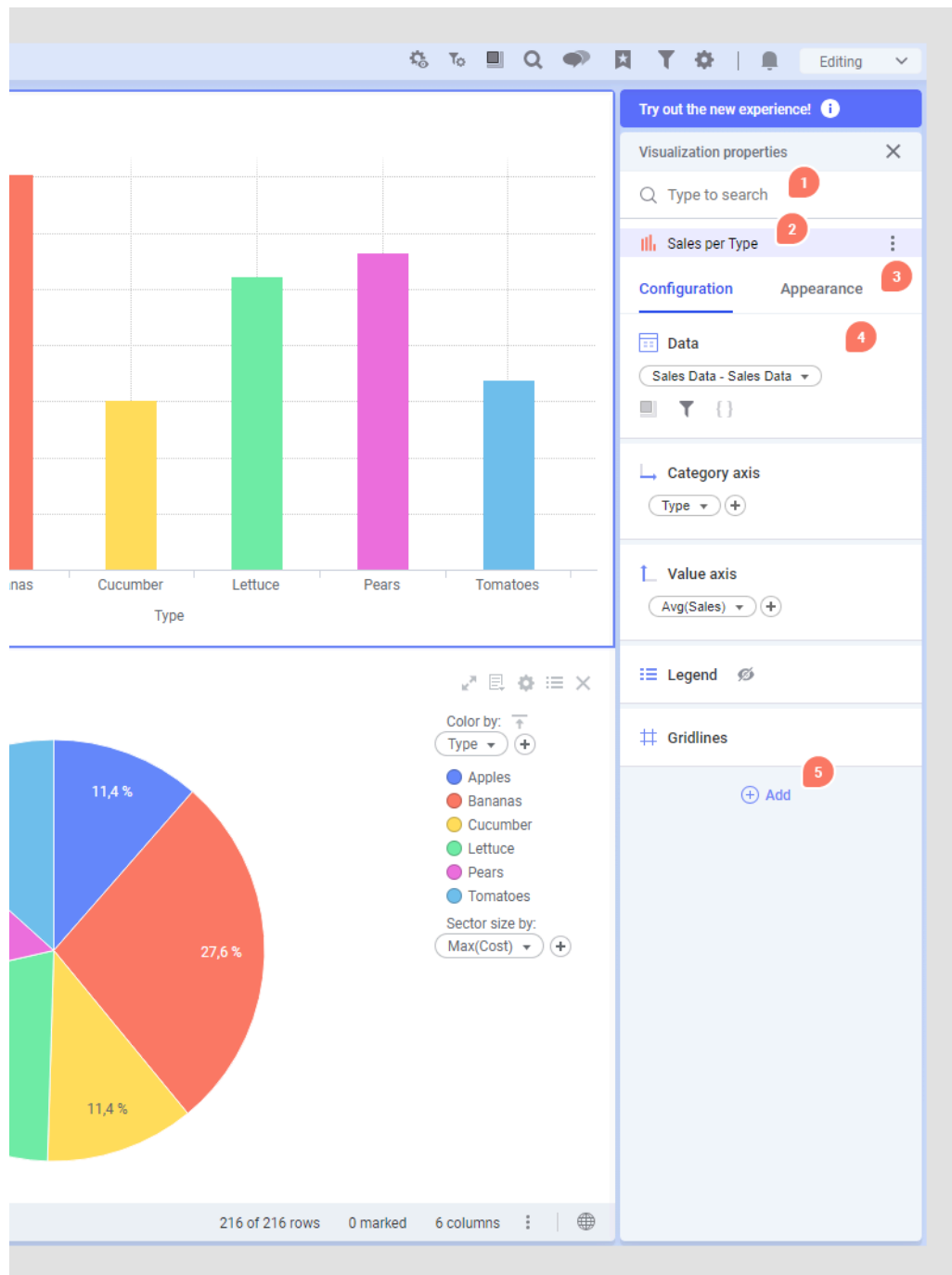
## Visualization properties (preview)

Some visualization properties can be accessed from the new visualization properties panel. To open the panel, select **View > Visualization properties (preview)**.




The redesign of the visualization properties is an ongoing process, and if you want, you can try out the new experience. For more information about the design goals and the benefits of the new experience, see [The new visualization properties experience in Spotfire](#) on the Community.

### The panel



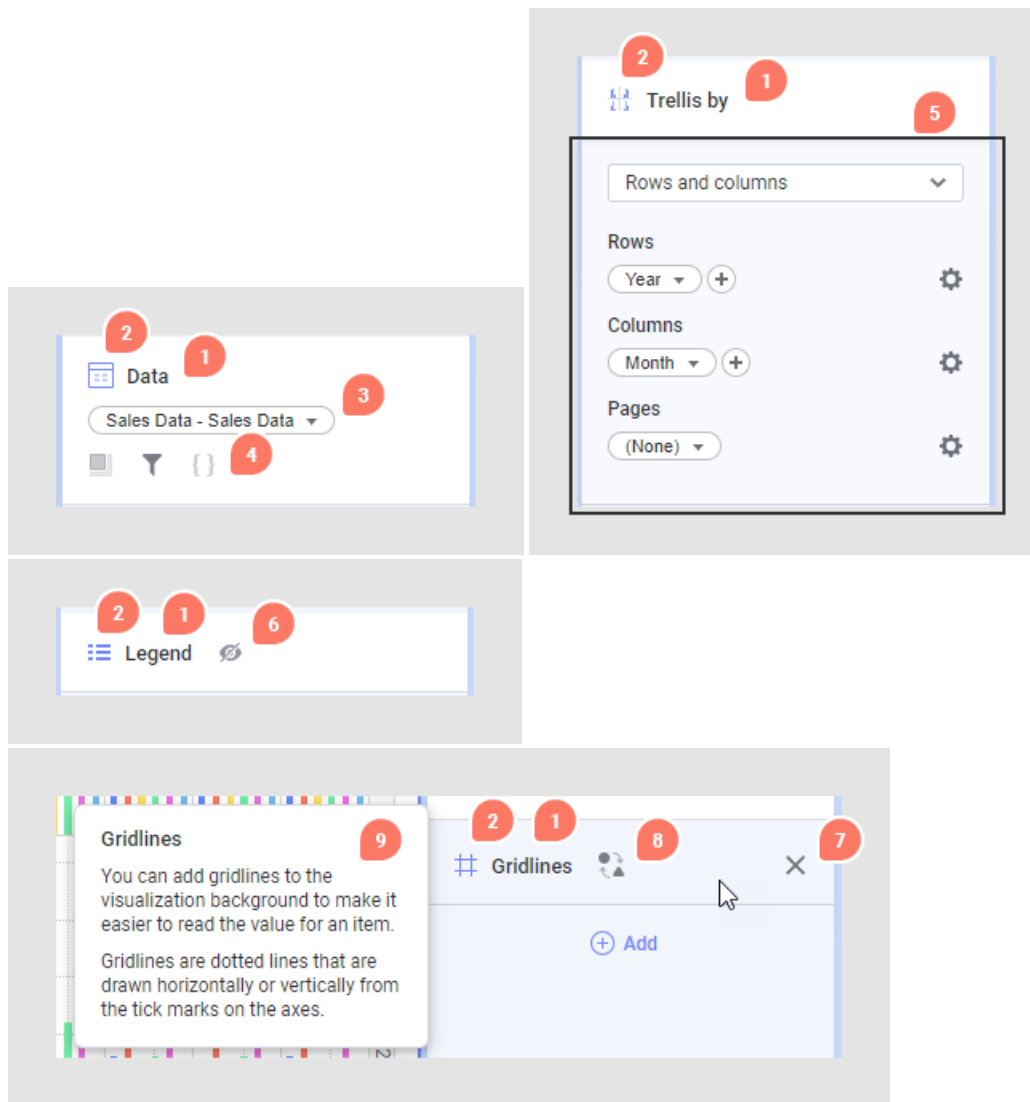
1. **Search bar** - Use the search bar to find properties and settings that have been configured for the selected visualization or visualizations. The search results also show all properties that are applicable to all selected visualizations and that you can add to the panel.
2. **Visualization titles, layers, dynamic items, etc.** - The title of the selected visualizations. Some visualizations can contain several subsections, such as layers (map chart) or dynamic items (graphical table).
3. **Tabs** - The properties are split into two categories, Configuration, and Appearance.
4. **Property cards** - Each visualization property and its related settings are usually contained in one property card. You can only see those cards in the panel that have been configured for the active visualization or visualizations. For more details, see the section below.
5. **Add** - Use the Add menu to find and add property cards to the panel that are applicable to the selected visualization or visualizations and that have not been configured yet.

### The properties cards

Each property card represents one visualization property or setting. As soon as a property is configured, its card is shown in the panel. To configure a property and thus add it to the panel, click  **Add** and select the property you want to apply to the selected visualization.

The property cards can differ in their behavior and how they can be used. Some cannot be removed and are always configured and present in the panel. If you removed one of these property cards you would break the visualization, for example the Data card or the axes cards (X-axis and Y-axis). Most cards can be expanded to show more settings. Some settings are already accessible in the collapsed state. Other cards are either "on" or "off" and are applied to the visualization immediately. Adding them to the panel acts like selecting a check box or turning on a switch.





1. **Card name** - Each property has a card name. Some cards have a fixed, type-based name, such as X-axis or Category axis, and some have a dynamic one, for example, Marking.
2. **Icon** - A descriptive icon to quickly locate and distinguish properties. Most of the icons are static but some are dynamic. For example, the Marking icon switches color based on the selected color.
3. **Column selector or data table selector (some cards only)** - A column selector lets you switch the column used in the setting. In the Data card, the selector switches the data table.
4. **Data limitations (Data card only)** - Icons that indicate which data limiting is applied.
5. **Settings of property** - This section becomes visible when expanding the card and it contains the settings.
6. **Hide** - For some cards, selecting Hide from the context menu toggles the visibility of certain properties in the visualization without removing the actual configuration or removing the card from the panel. For example, you can hide the Legend from the selected visualization.
7. **Remove** - When you remove a card from the panel, you remove it from the visualization and clear all configurations.
8. **Synchronize settings** - Synchronize the settings between all selected visualizations and choose which visualization's settings should be applied to all selected visualizations.
9. **Help tooltip** - Hover over a card or a setting to show a help tooltip that contains useful information.



If you selected more than one visualization, all properties that are applicable to all selected visualizations are shown in the panel. If settings differ this is indicated in the cards and you can sync the settings.

### Switching off the Visualization properties panel

The panel can be switched off by an administrator, using the `EnableVisualizationPropertiesPanel` preference in the Administration Manager under **Application > ApplicationPreferences**.

# Working with visualizations

---

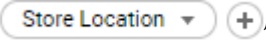
When a new visualization is created, a suggestion of what to visualize is created by default. You can then modify the visualization to show the values you are interested in by changing the columns shown on various axes, changing the aggregation, the coloring, etc.

Columns and aggregations are changed using the column selectors directly in the visualization, whereas appearance settings are specified using the visualization properties. Right-click the visualization, and select **Properties** in the opened menu, to change properties.

## Selecting columns on an axis


---

When creating a visualization, you select which data columns to show on various axes such as on the X-axis, the Y-axis and the color axis.

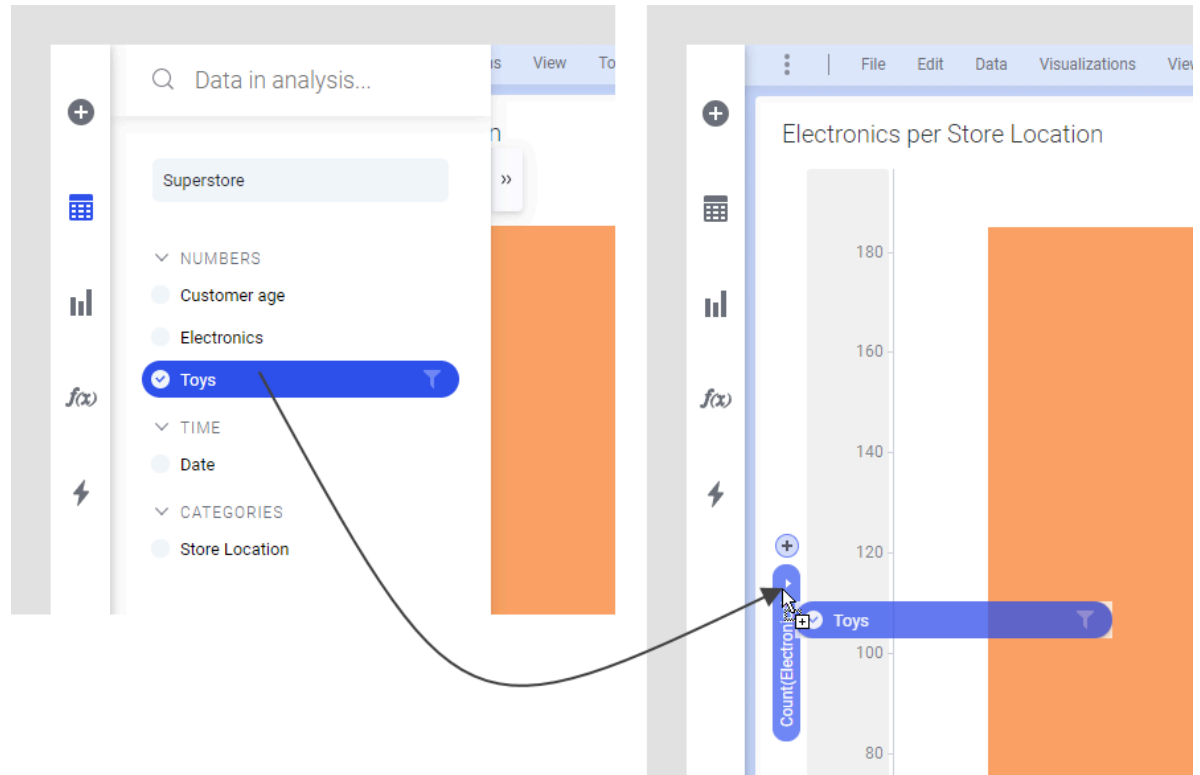
When hovering with the mouse pointer over a visualization, you find column selectors along the axes, for example , which displays the current column. To switch to another column, or add more columns to an axis, either use the **Data in analysis** flyout, or use the column selector itself.

## Selecting columns using the Data in analysis flyout

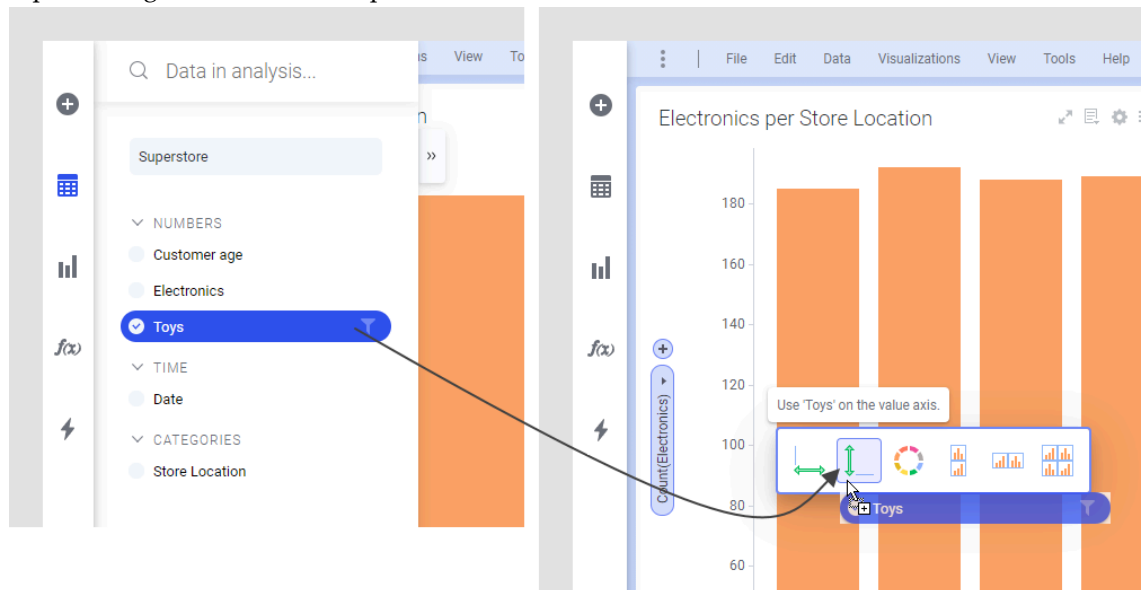
### Procedure

1. If the **Data in analysis** flyout is not visible, click **Data in analysis**  on the authoring bar.

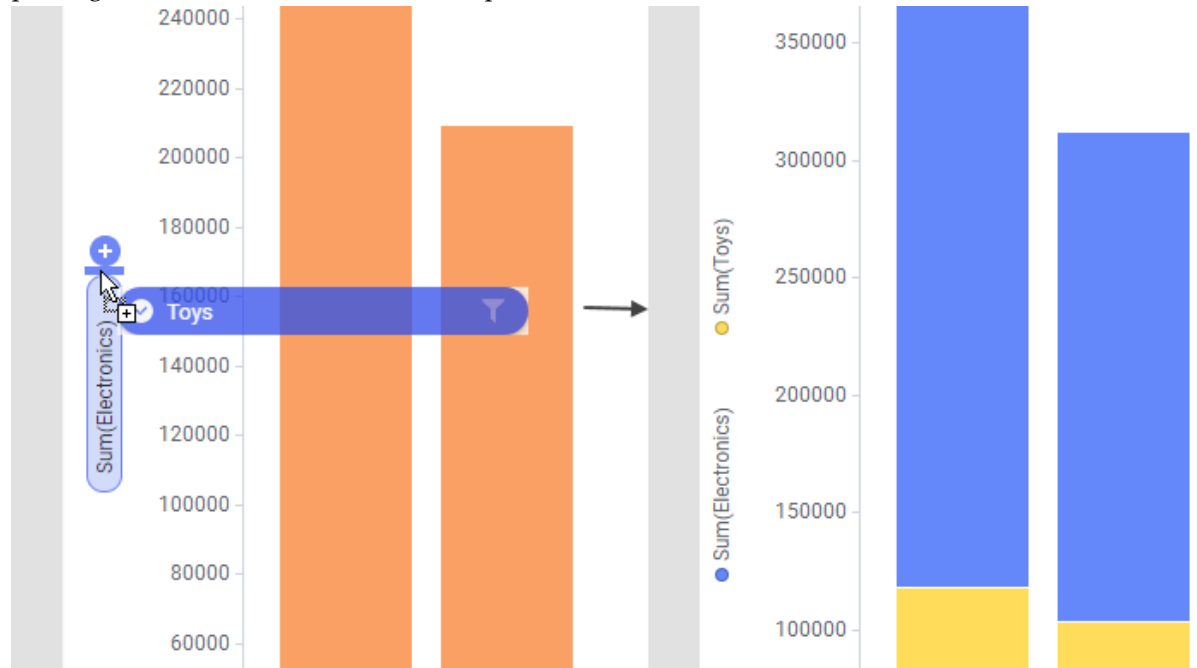
2. In the flyout, find the column you want to visualize and drag it to the column selector in question.



- Alternatively, drag the column to the visualization, where different drop targets appear representing various axes. Drop the column on the wanted axis.



- To display more than one column on the axis, select another column in the flyout, and drag it to the plus sign next to the column selector. Repeat to add more columns.

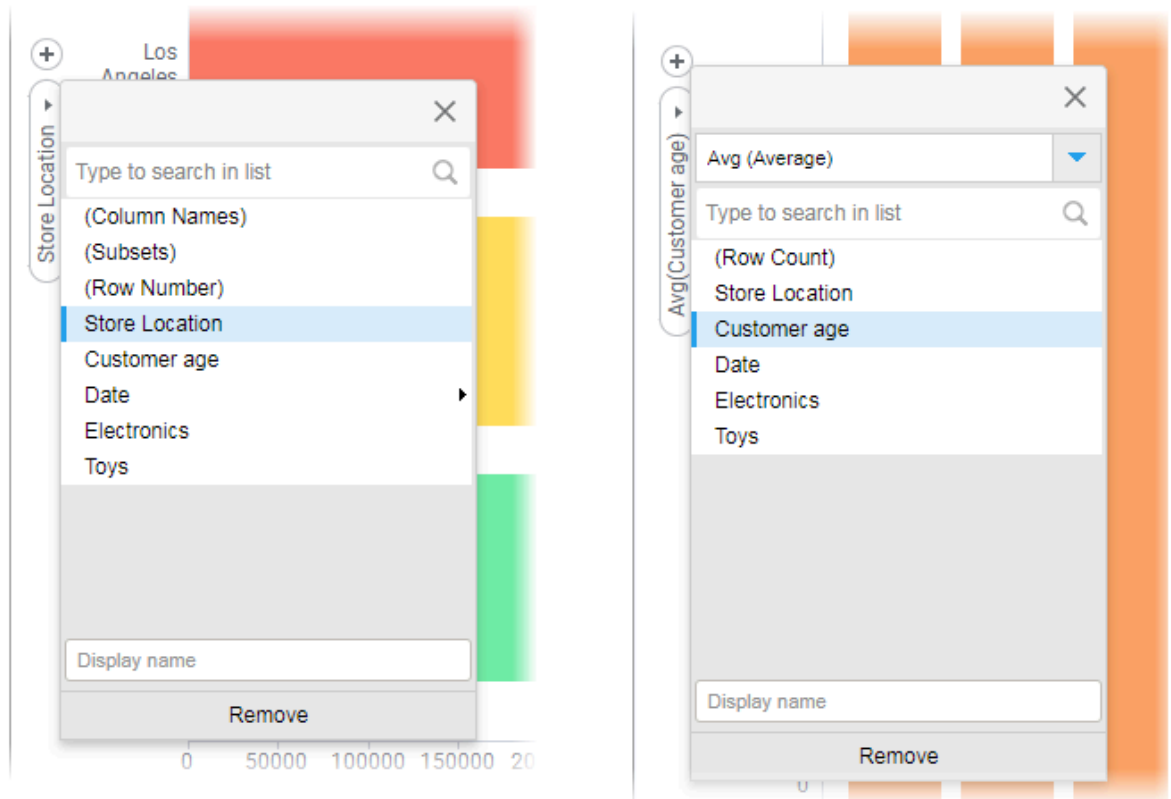





When the number of columns added to an axis exceeds three, the name of each column will no longer be shown, only the number of columns.

## Selecting columns using the column selector

### Procedure

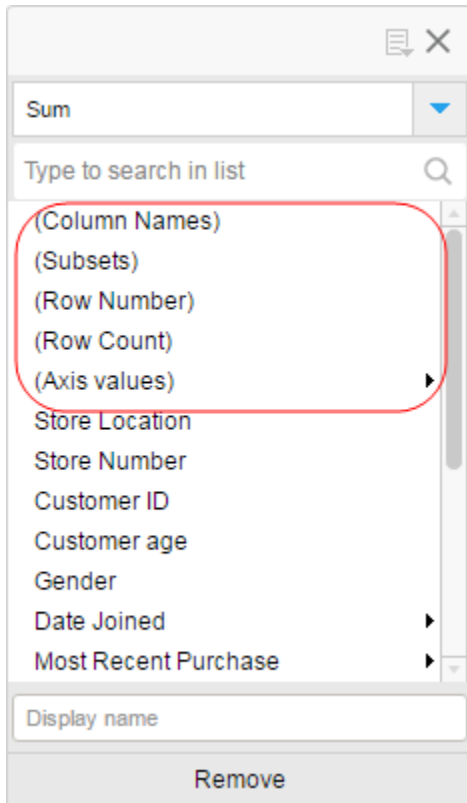
1. In the column selector, click the arrow next to the column name.  
A popover listing all columns in the data table opens. Depending on the type of axis and type of column data, the popover might look slightly different, as shown below. Axis selectors in the installed client can also be expanded to show more functionality.



2. Select a column in the popover.  
 When applicable, also select [aggregation](#) method.
3. If you want to display more than one column on the axis, click the plus sign next to the column selector, then select the column of interest.  
 In the **Display name** field, you can [specify your own display name for the axis](#). In the installed client you must expand the popover to see the field.
4. Click anywhere outside the popover to close it.  
 When the number of columns added to an axis exceeds three, the name of each column is no longer shown, only the number of columns.

## Other options than columns in column selectors

You can encounter other options than actual column names in the column selectors.

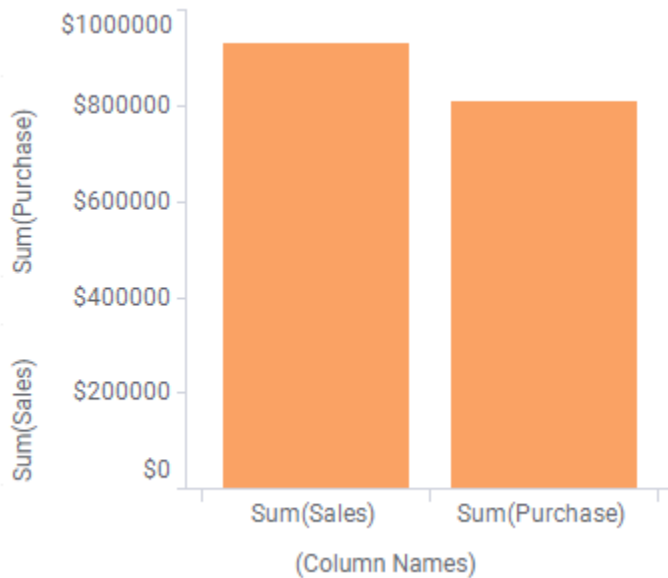


### (Column Names)

When more than one column is specified on a continuous axis in a visualization (such as the value axis in a bar chart), the option (Column Names) becomes available for selection in column selectors on other axes. The term (Column Names) represents the names of these columns. If (Column Names) is selected on an axis, the names of the columns will be treated as categories on the axis.

This is illustrated below where the totals of the Sales and Purchase columns in the data table are compared in a bar chart. When using (Column Names) on the category axis, the two columns selected on the value axis generate two categories on the category axis.

Transaction ID	Date	Sales	Purchase
SSMM55001	7/21/2017	6999	5165
SSMM55002	7/23/2017	7156	4449
SSMM55003	7/23/2017	731	1369
SSMM55004	7/23/2017	1431	1419
SSMM55005	7/23/2017	5429	5394
SSMM55006	7/23/2017	467	4286
SSMM55007	7/23/2017	467	122
SSMM55008	7/23/2017	0	684
SSMM55009	7/23/2017	0	0
SSMM55010	7/24/2017	747	0
SSMM55011	7/25/2017	405	153
SSMM55012	7/25/2017	2439	251
SSMM55013	7/26/2017	0	0



Any changes in the column selectors on the value axis above will automatically be reflected in the (Column Names) setting.



If you wish, you can color the bars differently by selecting (Column Names) also on the Color by axis.



You can [change what is displayed on the axes](#). For example, if you on the column selector above change Sum(Sales) to Sales , the display name of the bar will also be changed from Sum(Sales) to Sales.

## (Subsets)

When you have filtered data in your analysis, it is possible to display the following three categories; the data that is filtered to, the data that is filtered out, and the entire data. These categories can be compared in a single visualization by selecting the (Subsets) option in a column selector.

Which categories to compare in a visualization is specified in its Properties popover, the **Compare subsets** section.

Compare subsets

☐ All data
   
☐ Not in current filtering
   
☒ Current filtering

By selecting combinations of the check boxes, you can in a single visualization compare:

- The data in **Current filtering** (the data that has been filtered to) and **All data**
- The data in **Current filtering** and the data that is **Not in current filtering** (the data that has been filtered out)
- The data that is **Not in current filtering** and **All data**
- All three categories



To put the settings in effect, you select the (Subsets) option in the column selector of interest.



Because the usual behavior in visualizations is that they should reflect the [filtered data](#), the **Current filtering** check box is selected by default.

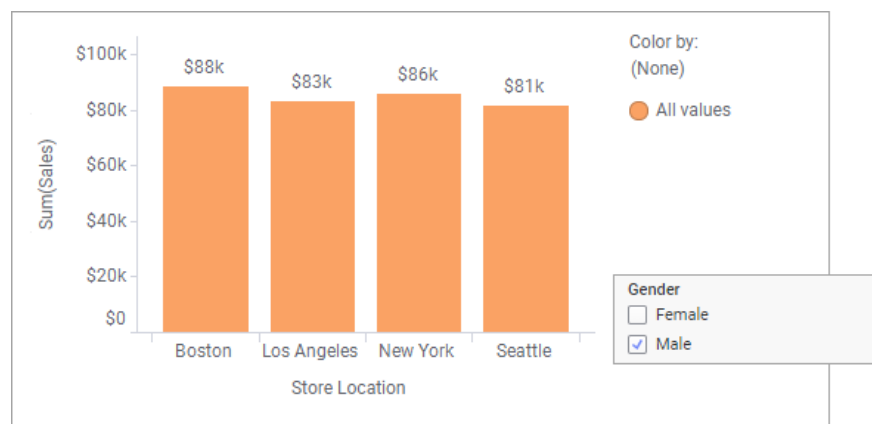
### Example of comparing filtered data to all data

Store Location	Sales	Gender
Boston	\$885	Male
Los Angeles	\$153	Male
New York	\$260	Male
Seattle	\$5k	Male
Seattle	\$59	Male
Boston	\$1k	Male
Seattle	\$7k	Male
New York	\$916	Male
Los Angeles	\$81	Male
New York	\$75	Male
Los Angeles	\$587	Male
Seattle	\$3k	Male
Los Angeles	\$555	Male

The bar chart is based on the data table above, and it shows the total sales for stores in four cities.



Assume you are interested in examining how much was spent by men only, and therefore filter to Male in the Gender column. The result is shown below:

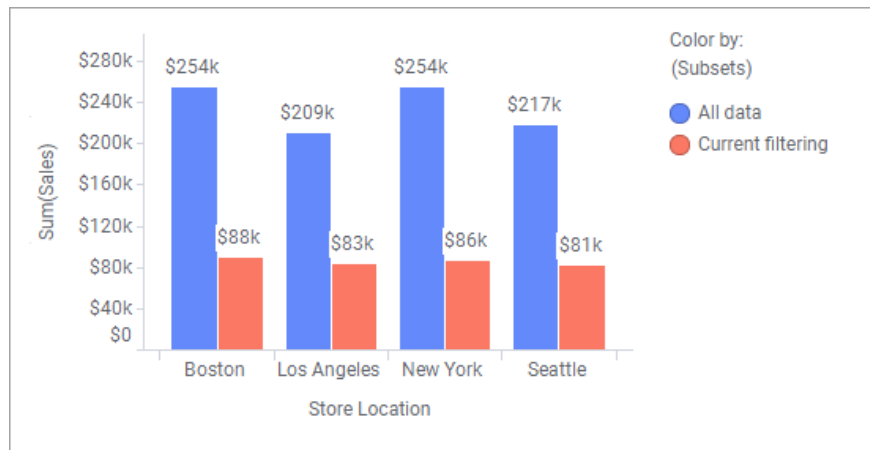


Using subsets, you can display the eight bars above simultaneously in the bar chart by first selecting the **All data** and the **Current filtering** check boxes in the Properties popover:

Compare subsets

☒ All data  
☐ Not in current filtering  
☒ Current filtering

Then select (Subsets) for example on the **Color by** axis, and you will be able to compare the two categories in the bar chart.



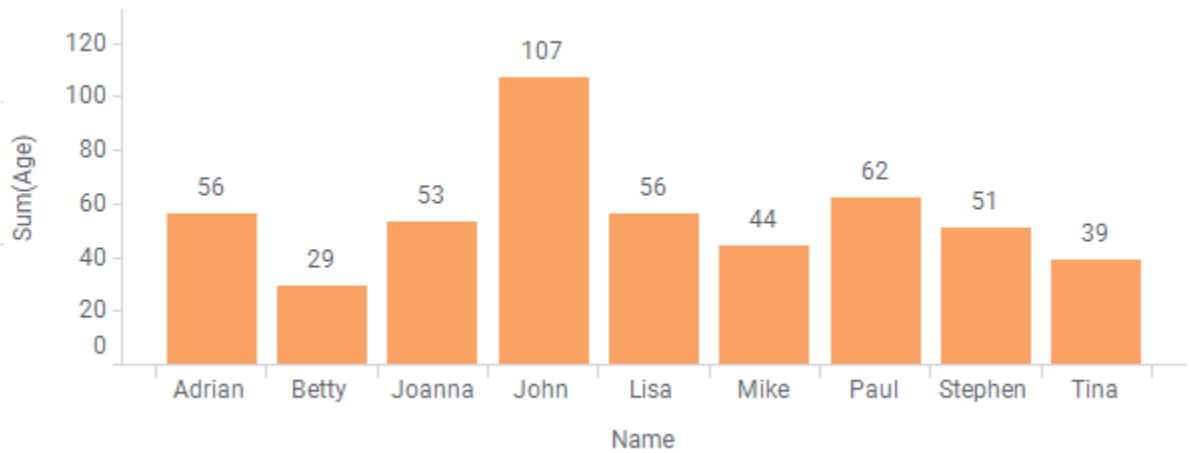
### (Row Number)

You select the (Row Number) option in a column selector, when you want each row in the data table to be shown individually in a visualization.

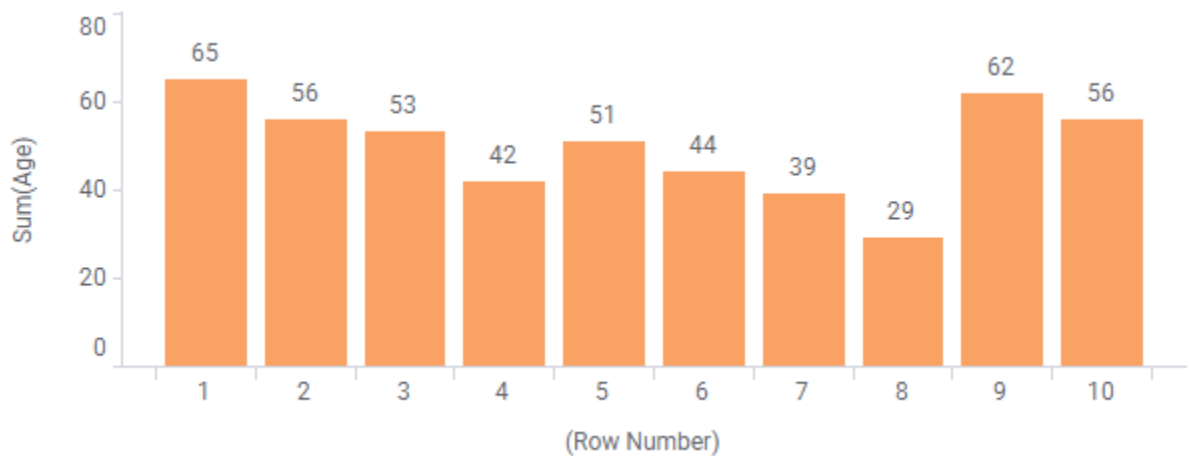
#### Example

Gender	Name	Age
Male	John	65
Female	Lisa	56
Female	Joanna	53
Male	John	42
Male	Stephen	51
Male	Mike	44
Female	Tina	39
Female	Betty	29
Male	Paul	62
Male	Adrian	56

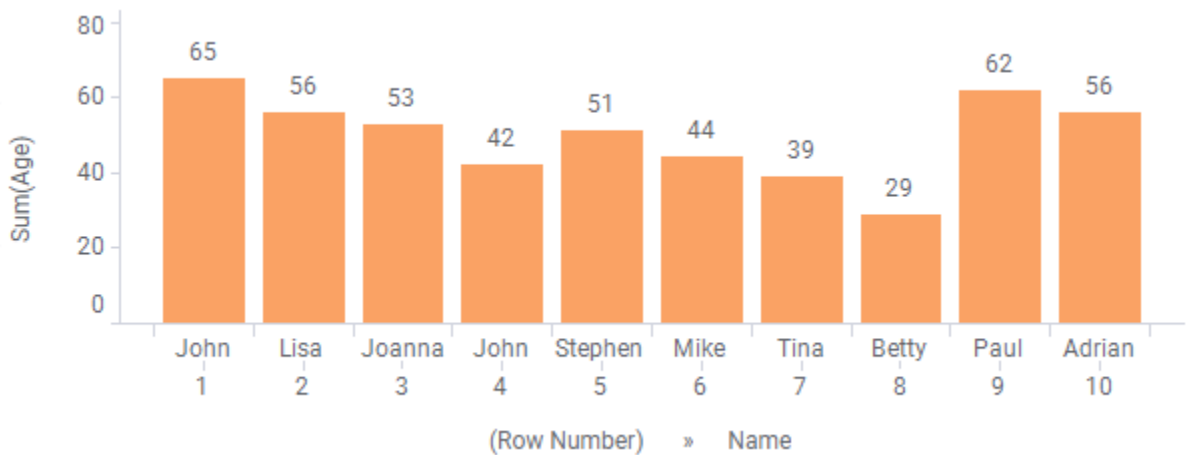
Assume you want to show the age of each person in the data table above as bars in a bar chart. If you select 'Name' on the category axis and do not notice there are two Johns, the ages of these two persons will be aggregated as shown below.



Selecting (Row Number) instead of 'Name' on the category axis will present each row in the data table individually. The value 'John' is listed in row 1 as well as in row 4 in the data table, but the value is now represented as two individual bars, '1' and '4'.



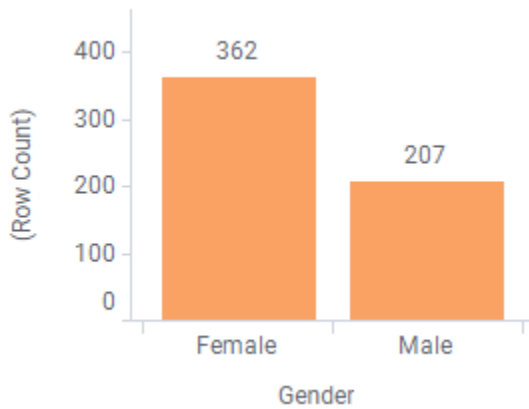
If it is important to show the actual names, select both (Row Number) and 'Name' on the category axis.



## (Row Count)

You select the (Row Count) option in column selectors when you want to display how the rows are distributed between different categories in a data column.

For example, the data table used in the bar chart below contains a 'Gender' column, whose values are Female or Male. If you select (Row Count) on the Value axis, the bars display how many were women and how many were men in the 'Gender' column.



If you want [another display name](#) than (Row Count) on the axis, you can change it.

Another example of using the (Row Count) option is shown below. The cross table is based on a data table containing sales figures for fruit, spices, and vegetables in four different regions. Using the (Row Count) setting on the Cell values axis, the cross table displays the number of sales transactions made per category and region. For example, 834 of the data table rows were fruit transactions in the Midwest region.

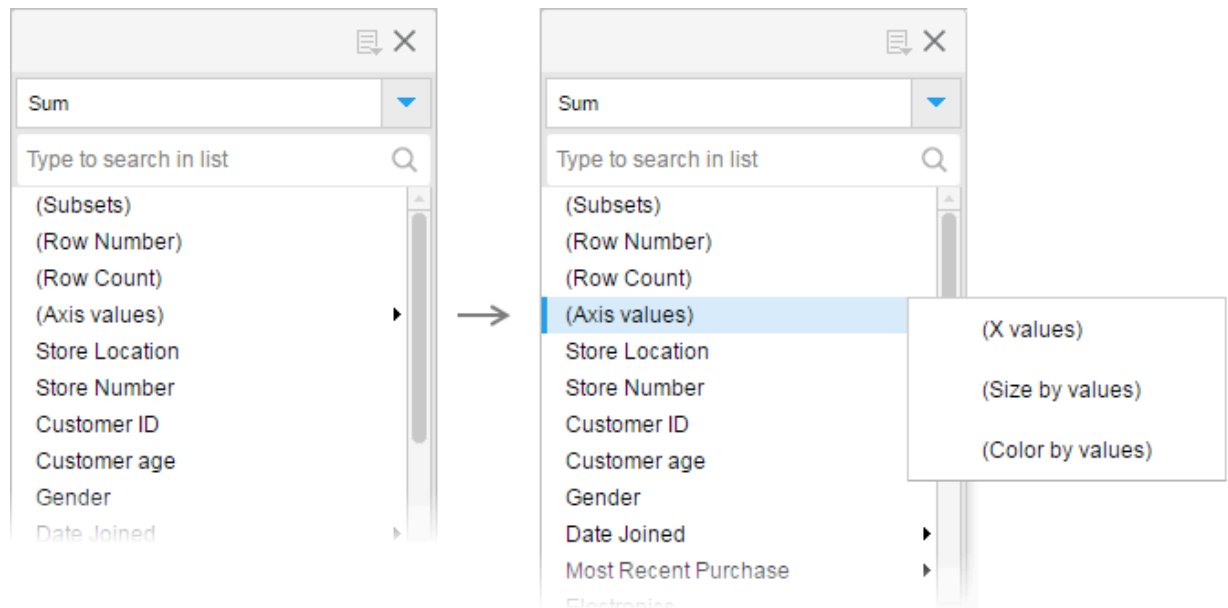
	Region				Colors:
	Midwest	Northeast	South	West	
Category	834	763	1404	1132	● Max
Fruit	430	402	832	584	● Min
Spices	150	127	243	184	
Vegetables					

(Row Count)

Furthermore, the result from the (Row Count) setting is used indirectly for coloring the cross table cells. This is because the Color by axis in the Properties popover is set to (Cell values).

## (Axis values)

In column selector menus, you can encounter different (Axis values) options, for example, (Value axis values), (Color by values), and (X values). Such a non-column option represents the resulting values on the axis in question. It is especially useful for referring to axes with multiple columns.



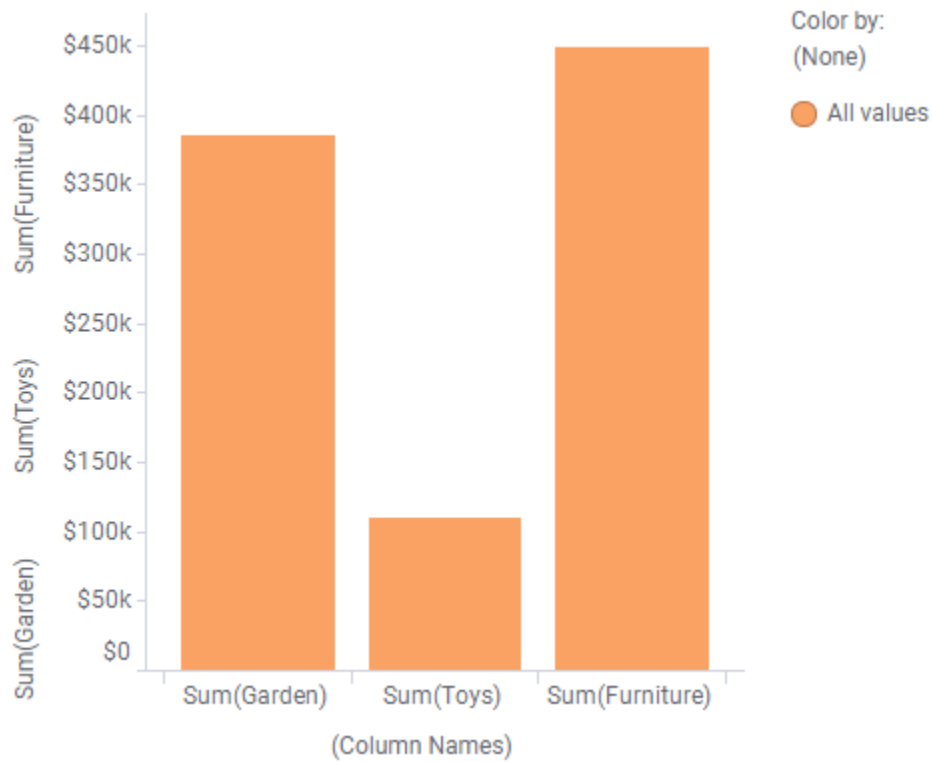
Which options are available in a column selector depends on the visualization type and what is selected on the visualization axes. If there are several axis values options, they will be displayed by clicking the (Axis values) arrow as shown above. If only one option exists, it will be available directly.



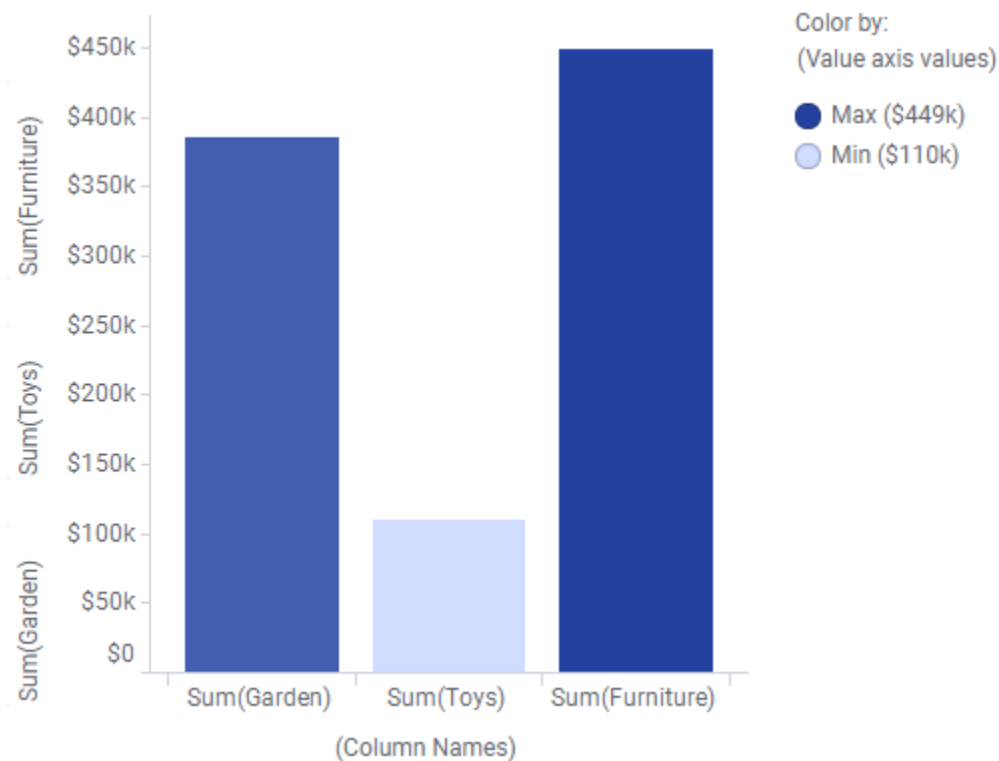
In a [waterfall chart](#), (Value axis values) is the default selection on the color by axis.

## Example

The bar chart shows the sums of sales at three different departments in a store. The data is retrieved from three columns, that is, more than one column is selected on the value axis.



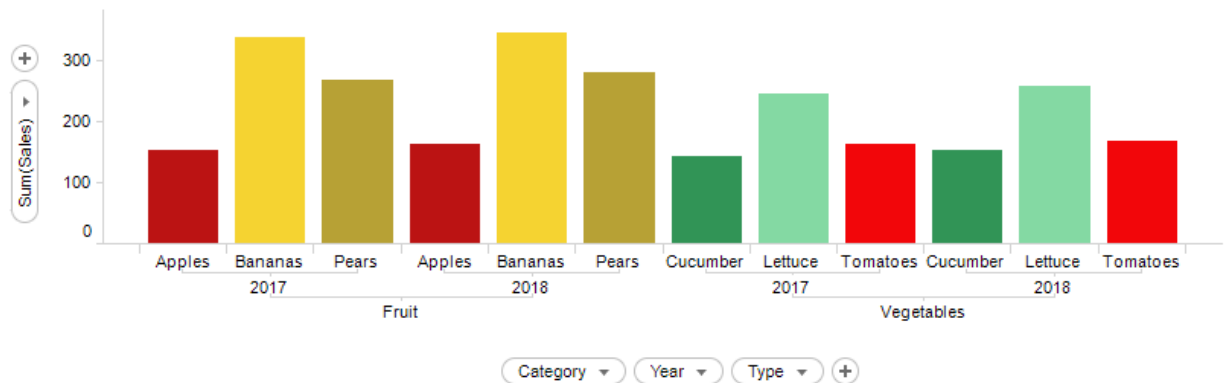
By selecting (Value axis values) on the Color by axis, all bar values on the value axis become part of the coloring scale as shown below. This means you can use a single gradient scale when coloring the bars, even though they represent values from multiple columns.



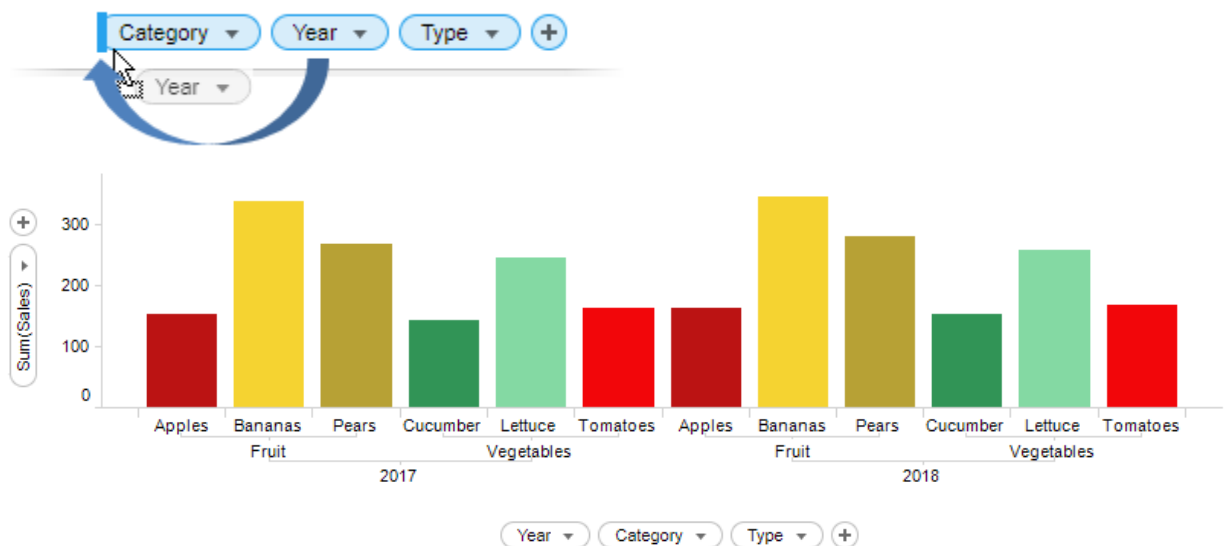
If you change what is selected on the value axis above, the coloring automatically updates to the new selection. This is because of the (Value axis values) selection on the color axis.

## Changing the column order on an axis

It is possible to change the order of columns on an axis by dragging the columns to other positions. For example, in the bar chart below, three columns are selected on the category axis.



If you for example drag the Year column to the left, the bar chart changes accordingly as shown below.



### Prerequisites

More than one column are selected on an axis.

### Procedure

- Specify the order of the columns on the axis by dragging them to the wanted positions.

## Drag-and-Drop

Spotfire contains rich possibilities of using drag-and-drop operations for setting up the visualizations. You can drag columns from the data in analysis flyout, filters from the filters panel, or even column selectors and drop them on the visualization axes and drop targets in the middle of the visualizations. These drop targets control coloring, trellising, size or shape, etc. You can also [change the layout of visualizations on the pages](#) using drag-and-drop. All operations can be undone, so you can try different layouts without being afraid of destroying anything.



## Column selectors

Dropping a column (a filter or a column selector) directly on a column selector will change that axis to use the new column instead. Dropping it next to the previous column selector will add another column to the axis.

## Copying pages and visualizations

You can drag a page or a visualization from one instance of Spotfire to another if you want to reuse the page structure or visualization set-up in another analysis.
























No data is included when a page or visualization is copied to a new instance of Spotfire. If the new data does not match the previously used data, you must specify the data tables and columns to use manually.












## Drop targets

You can drag a filter from the filters panel or a column from the Data in analysis flyout and move it over the center of a visualization. As you do so, drop targets will show up as seen in the table below. By dropping a filter on, for example, the **Color by** icon, the visualization will be colored according to the column the filter represents. Other drop targets can be used to set different properties, such as X-axis or Trellis. If you want to assign several columns to one property, you can use **Shift + Click** or **Ctrl + Click** to select several filters at once.

You can also drag a column selector from, for example, the legend or an axis and drop that on a drop target. The behavior is identical to that of dragging a filter or column except that the original column selector will be removed unless you hold down **Ctrl** while dragging. This behavior is the same if you drag the columns directly to a column selector.

Drop target	Visualization	Description
	Bar chart, scatter plot, line chart, parallel coordinate plot, box plot, combination chart, waterfall chart	Changes column on the horizontal axis. In most cases this means the X-axis in a visualization, but in a bar chart it depends on the orientation of the bars. In a vertical bar chart, this refers to the category axis; in a horizontal bar chart it refers to the value axis.
	Bar chart, scatter plot, line chart, box plot, combination chart, waterfall chart	Changes column on the vertical axis. In most cases this means the Y-axis of a visualization, but in a bar chart it depends on the orientation of the bars. In a vertical bar chart, this refers to the value axis; in a horizontal bar chart it refers to the category axis.
	3D scatter plot	Changes column on the X-axis in the 3D scatter plot.
	3D scatter plot	Changes column on the Y-axis in the 3D scatter plot.
	3D scatter plot	Changes column on the Z-axis in the 3D scatter plot.
	Bar chart, line chart, parallel coordinate plot, pie chart, scatter plot, 3D scatter plot, map chart, box plot, treemap, waterfall chart, KPI chart	Colors the visualization items by the provided column.











Drop target	Visualization	Description
	Scatter plot, 3D scatter plot, map chart	Changes the markers' shapes according to the values in the provided column.
	Scatter plot, 3D scatter plot, map chart	Sizes the markers by the provided column.
	Line chart	Splits the lines according to the values in the provided column, unless they have already been split by a more detailed coloring option.
	Pie chart	Defines the column or hierarchy whose values will set the size of the pie sectors.
	Treemap	Changes the treemap hierarchy to the provided column or hierarchy.
	Treemap	Sizes and orders the rectangles in the treemap by the provided column.
	Cross table	Changes the cell values of the cross table to the provided column.
	Cross table, graphical table, heat map	Changes the vertical values (Y-axis) to the provided column.
	Cross table, heat map	Changes the horizontal values (X-axis) to the provided column.
	Heat map	Changes the cell values of the heat map to the provided column.
	Heat map	Adds the provided column to the cell values of the heat map.
	Table, graphical table, summary table	Shows the selected columns in the table (and removes all other columns from the table).
	Table	Adds the selected columns to the table.
	Summary table	Splits the summary table into different categories.
	Bar chart, line chart, parallel coordinate plot, pie chart, scatter plot, 3D scatter plot, map chart, box plot, treemap, combination chart, heat map, waterfall chart	Splits (trellises) the visualization into several rows, where the number of panels equals the number of categories in the column provided.

Drop target	Visualization	Description
	Bar chart, line chart, parallel coordinate plot, pie chart, scatter plot, 3D scatter plot, map chart, box plot, treemap, combination chart, heat map, waterfall chart	Splits (trellises) the visualization into several columns, where the number of panels equals the number of categories in the column provided.
	Bar chart, line chart, parallel coordinate plot, pie chart, scatter plot, 3D scatter plot, map chart, box plot, treemap, combination chart, heat map, waterfall chart	Splits (trellises) the visualization into several panels, where the number of panels equals the number of categories in the column provided.
	Bar chart, line chart, parallel coordinate plot, pie chart, scatter plot, 3D scatter plot, map chart, box plot, treemap, combination chart, heat map, waterfall chart	(Only available when a column selector within a visualization is being dragged.)  Removes a previously applied condition from the visualization. For example, if you have dragged a filter to the "Color by" drop target of a visualization, but later want to remove the coloring, you can drag the color selector from the legend to this drop target.
	KPI chart	Changes column on the Value axis in a KPI, that is, changes what is displayed as the primary value in a tile.
	KPI chart	Changes column on the Time axis in a KPI, that is, changes what is defined as the time period to base the KPI values on.
	KPI chart	Changes column on the Tile by axis in a KPI, that is, splits the KPI into tiles according to the values in the column.
	KPI chart	Changes column on the Comparative value axis in a KPI, that is, changes the column that the primary value should be evaluated against.
	KPI chart	Adds a new KPI to the KPI chart. The selected column is applied on the Value axis in the KPI.
	All visualizations except graphical table	Attaches the selected tag to the marked rows.
	Map chart	Adds a new marker layer to the map chart based on the data table containing the selected column.
	Map chart	Adds a new marker layer to the map chart based on the data table containing the selected column.

### Graphical table specific drop targets


The [graphical table](#) has, depending on what dynamic items are shown in the table, a number of drop targets specific to the graphical table. If multiple items of the same type are available, you can always see which graphical table column the drop target belongs to by looking at the highlighted column in

the graphical table when hovering over a drop target. Note that the added columns must come from the same data table as the values on the row axis in the graphical table for anything to be displayed.

Drop target	Description
	Uses the selected column to specify the Y-axis values in the highlighted sparkline column.
	Uses the selected column to specify the X-axis values in the highlighted sparkline column.
	Uses the selected column to specify the calculated values in the highlighted calculated value column.
	Uses the selected column to specify the icons in the highlighted icon column.
	Uses the selected column to specify the values in the highlighted bullet graph column.
	Uses the selected column to specify the comparative values in the highlighted bullet graph column.
	Adds a new sparkline column to the graphical table, using the selected column on the Y-axis of the sparkline. The default aggregation for numerical columns is Sum. The default for categorical columns is Count.
	Adds a new calculated value column to the graphical table, using the selected column to calculate the values. The default aggregation for numerical columns is Sum. The default for categorical columns is Count.
	Adds a new icon column to the graphical table, using the selected column to calculate the icons. The default aggregation for numerical columns is Sum. The default for categorical columns is Count. The default rule for icons is Top 1.
	Adds a new bullet graph column to the graphical table, using the selected column to calculate the values. The default aggregation for numerical columns is Sum. The default for categorical columns is Count.

### Filters panel drop target

When you are working with in-database data or have turned off the automatic creation of filters in the Options dialog, you will see a drop target in the filters panel:

Drop target	Description
	(Only applicable for columns where no filter is previously available.) Creates a filter for the selected column.

### Column from marked

The **Column from marked** functionality is a way to look up which column to assign to a column selector by fetching the column name from a cell value in a data table, using marking. The cell would

then contain the name of the column you want to assign to an axis in a visualization, such as what to use on the X-axis or what to color by.

Right-click the axis selector of the axis of interest and select **Column from marked** from the opened menu.

You use the **Column from marked** functionality when you want to update the visualization iteratively and quickly set another column for the property. Once **Column from marked** has been configured, pointing to a new cell by marking a row updates the axis. If you only want to configure a visualization to use a certain column, you should not use the **Column from marked** functionality.

See [Column from marked - examples](#) for more information.



When the **Column from marked** functionality has been switched on, it will be used to continuously update the visualization until it is switched off. This means that any other configuration that you try to apply to the visualization will be lost each time the marking changes.

When configuring the functionality, you select:

- the data table containing the column with all column names
- the column itself
- the marking that should define what to show on the axis (if more than one item is marked, the first item will be used to define the value to use on the axis)

## Expression

By default, the expression used is the plain cell value as specified by the input fields at the top of the **Column from marked** dialog.

A suitable aggregation method is automatically selected depending on the type of axis in the visualization. The text "Sales 2022" can therefore be interpreted as `[Sales 2022]` on the Y-axis of a scatter plot, as `Sum([Sales 2022])` on the value axis of a bar chart and as `<[Sales 2022]>` on a categorical category axis of a bar chart.

However, if you want to have more control of how values should be interpreted, you can choose **Advanced**, and extend the default expression to include additional columns and/or apply aggregation methods.

The `{0}` variable refers to the value defined by the three input fields in the dialog (the plain cell value) and the cell value is automatically escaped to a column expression by the addition of straight brackets. The expression can be extended using the same [syntax](#) as custom expressions.

If additional columns are specified, these will show up as individual column selectors for the visualization property (axis) once the **Column from marked** functionality has been applied.

## Examples:

The expression `{0}, [Sales 2023]` applied on a continuous column containing the column names "Sales YearX" would result in two columns (where the second one always is Sales 2023) on the axis: `[Sales 2022]`, `[Sales 2023]` or `[Sales 2021]`, `[Sales 2023]`, etc., depending on what year was marked.

The expression `Sum({0})` would result in the expression `Sum([Sales 2022])` (with varying years depending on what you mark), hence, an aggregated, continuous column on the axis.

The expression `Sum({0}) over AllPrevious([Axis.X])` would result in the expression `Sum([Sales 2022]) over AllPrevious([Axis.X])`, hence, a cumulative sum on the axis.

The expression `<{0}>` used with a categorical column containing genders could result in the expression `<[Male]>`, `<[Female]>` or `<[Non-binary]>` on the axis, depending on what was marked.

The expression `<{0} nest [Product]>` with a categorical column containing the values On or Off would result in the categorical hierarchy `<[On] nest [Product]>` or `<[Off] nest [Product]>` on the axis, depending on what was marked.

## Using Column from marked - Examples

When you apply Column from marked on an axis in a visualization, you make it possible to switch the column on the axis's column selector by marking certain data in another visualization.

The two visualizations can be built on totally unrelated data tables. However, a prerequisite is that in the data table where you mark data, cell values in one of the data columns must be identical to column names in the other data table.

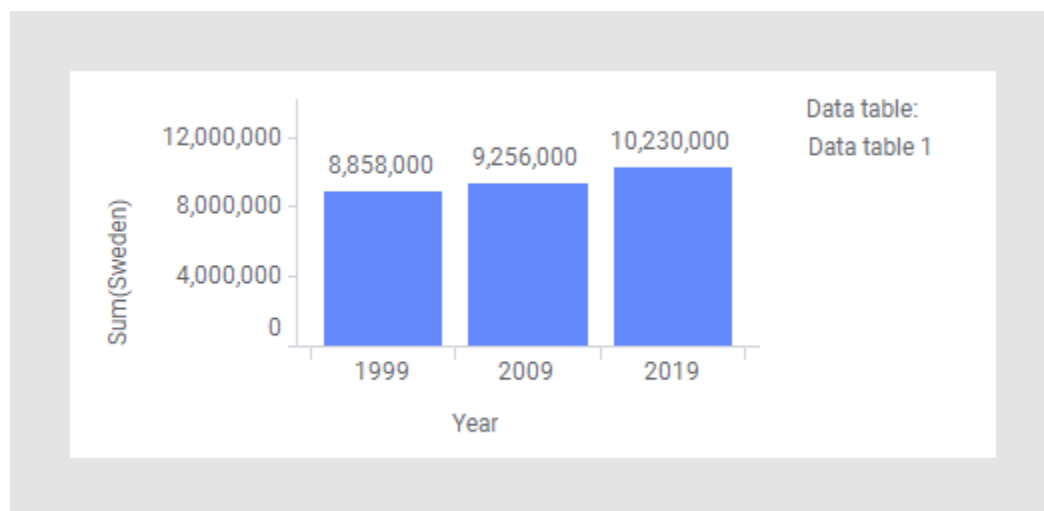
Two use cases of **Column from marked** follow below.

### Browse through different columns on an axis

An analysis contains two data tables, Data table 1 and Data table 2.

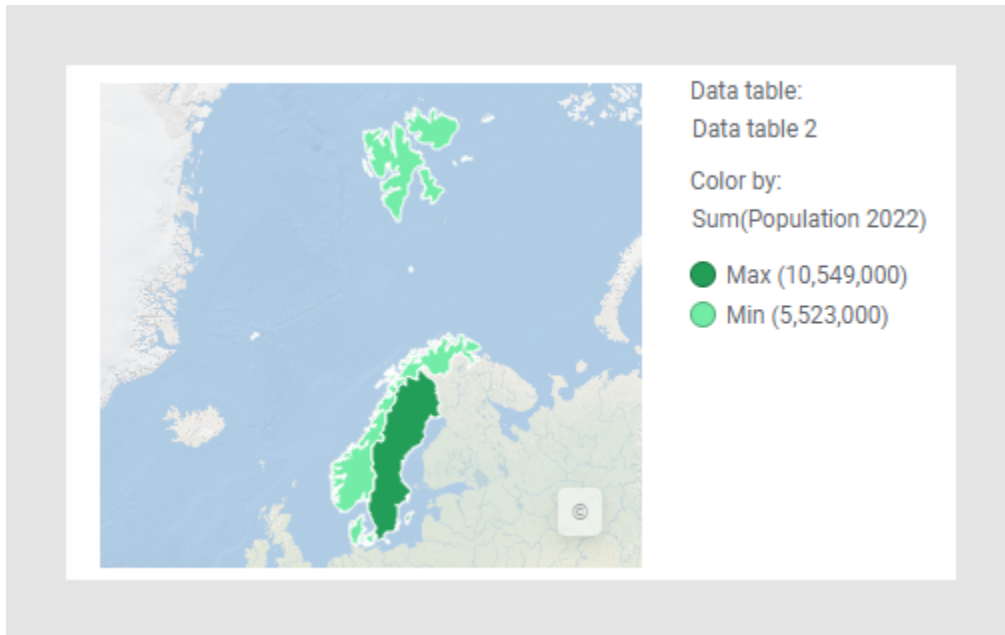
Data table 1 contains population data for Sweden, Denmark, and Norway for three years, and the bar chart below visualizes the yearly population for Sweden.

Year	Sweden	Denmark	Norway	Data table:
1999	8,858,000	5,322,000	4,462,000	Data table 1
2009	9,256,000	5,511,000	4,799,000	
2019	10,230,000	5,806,000	5,385,000	



Data table 2 lists the population 2022 for each country, and the feature layer in the map chart reflects the population numbers.

Country	Population 2022	Data table:
Sweden	10,549,000	Data table 2
Denmark	5,841,000	
Norway	5,523,000	



By applying **Column from marked** on the bar chart value axis, you can, by marking a country at a time in the map chart, show each country's yearly population in the bar chart. You right-click the value axis, and select **Column from marked** to open this dialog:

**Column from marked** ✕

**Set column name to cell value from this column and row.**

Data table  
Data table 2 ▼

Column  
Country ▼

Row from marking  
Marking ▼

**Expression**

☒ Default as specified above

☐ Advanced

{0}

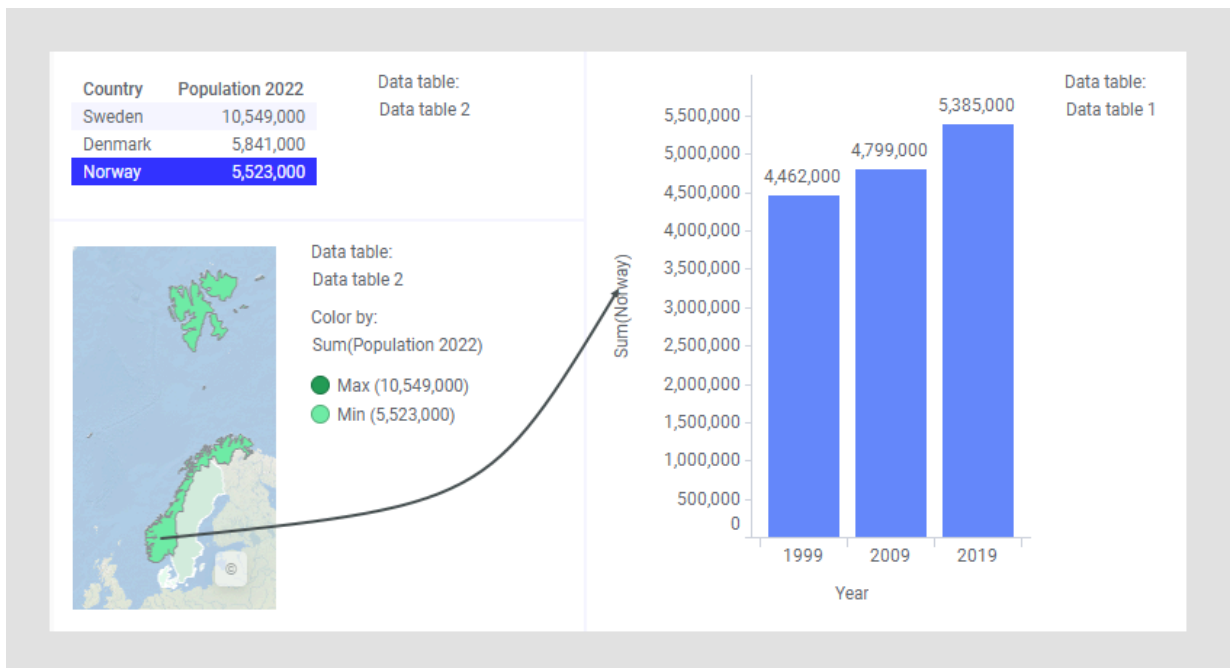
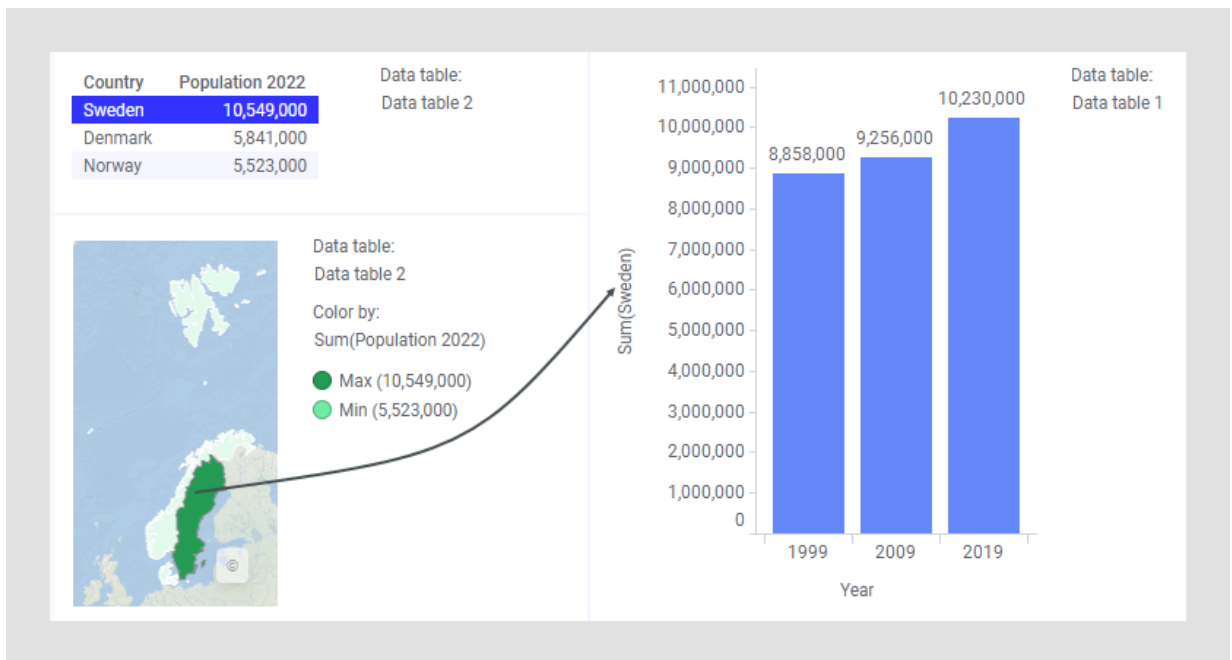
**Sample**  
(No marked rows)

Help OK Cancel

Select the data table (Data table 2), and the data column (Country), whose marked values are intended to control what is specified on the bar chart value axis.

Now you will be able to, for example, mark a country in the map chart, and the yearly population for that country will be displayed in the bar chart. See the images below where, firstly, 'Sweden' is clicked, and then 'Norway'.



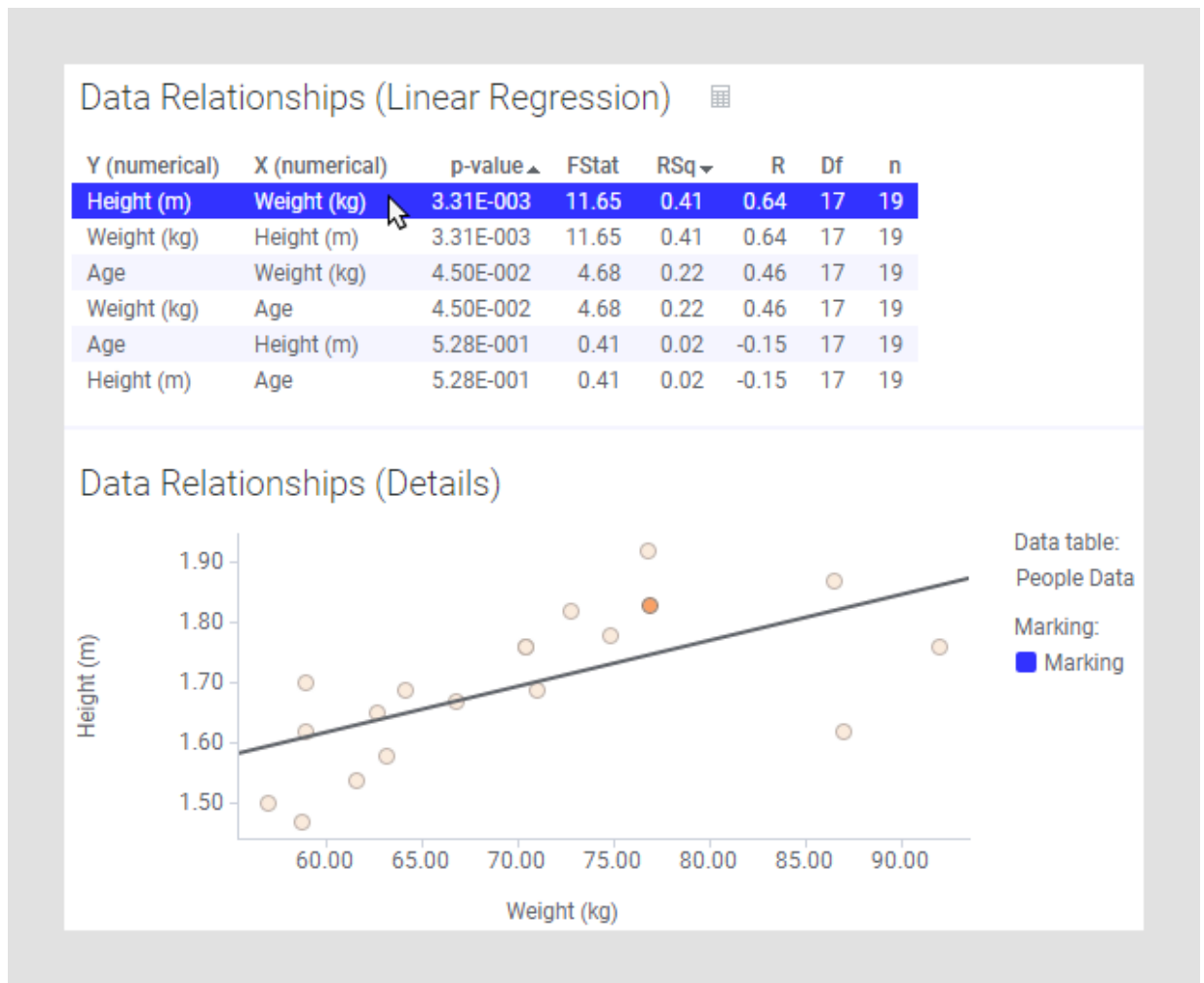


### Use Column from marked in Data Relationships

The **Data Relationships** tool (available in the installed client) uses **Column from marked** to display the results from the calculation. The purpose of **Data Relationships** is to calculate the correlation between the columns in a table, and then find and display highly correlated columns to see what they look like.

The result when using **Data Relationships** is a new data table, where each row shows the correlation between two columns from the original data table. The row includes the names of the two compared columns, and values for how well they correlate.

In addition to the table visualization that displays the new correlation data, the tool creates, in the linear regression case, a scatter plot that shows one column plotted against the other from the original data table as shown below.



As you mark a row in the new correlation data table, the scatter plot is updated to display the two compared columns from that row. In this configuration, the X- and Y-axes of the scatter plot are configured using **Column from marked**. That is, the names of which columns to display are fetched from the marked row in the new correlation data table.

You change the marking by clicking another row in the table visualization.



The way this is set up, you can quickly browse the correlation between columns by stepping through the rows in the table visualization. To update the scatter plot manually by interacting with the column selectors of the X- and Y-axes would take much longer.

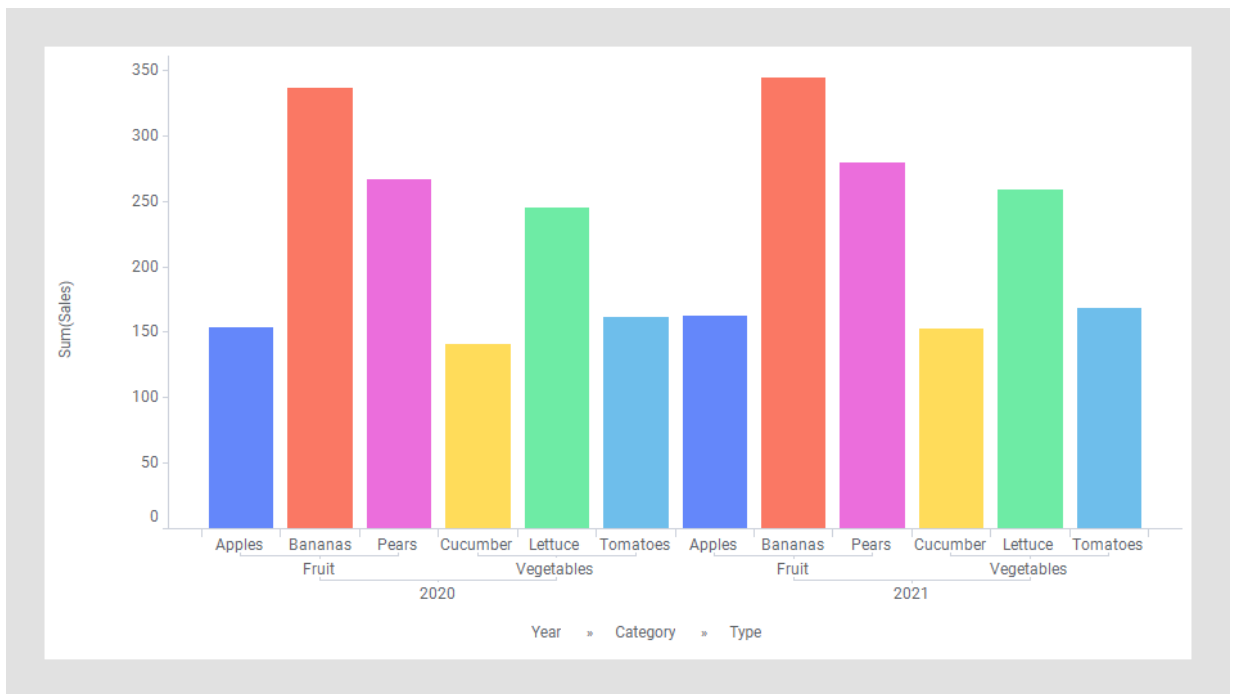
## Hierarchies

Data can be organized in a specific order where each value is either above or below in relation to the other values. Usually in this hierarchical structure, the most general value is at the top and the most detailed one is at the bottom.

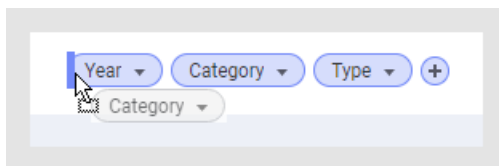
Hierarchical structures are often visualized in the form of tree diagrams, treemaps, or bar charts. Typically, product categories and the products therein, or geographical data like continent/country/city, et cetera, are suitable for using hierarchies.

In Spotfire, you can structure your data hierarchically by visualizing more than one column on an axis. In this way, you get a better overview of your data, can analyze it in more detail, and get better insights.

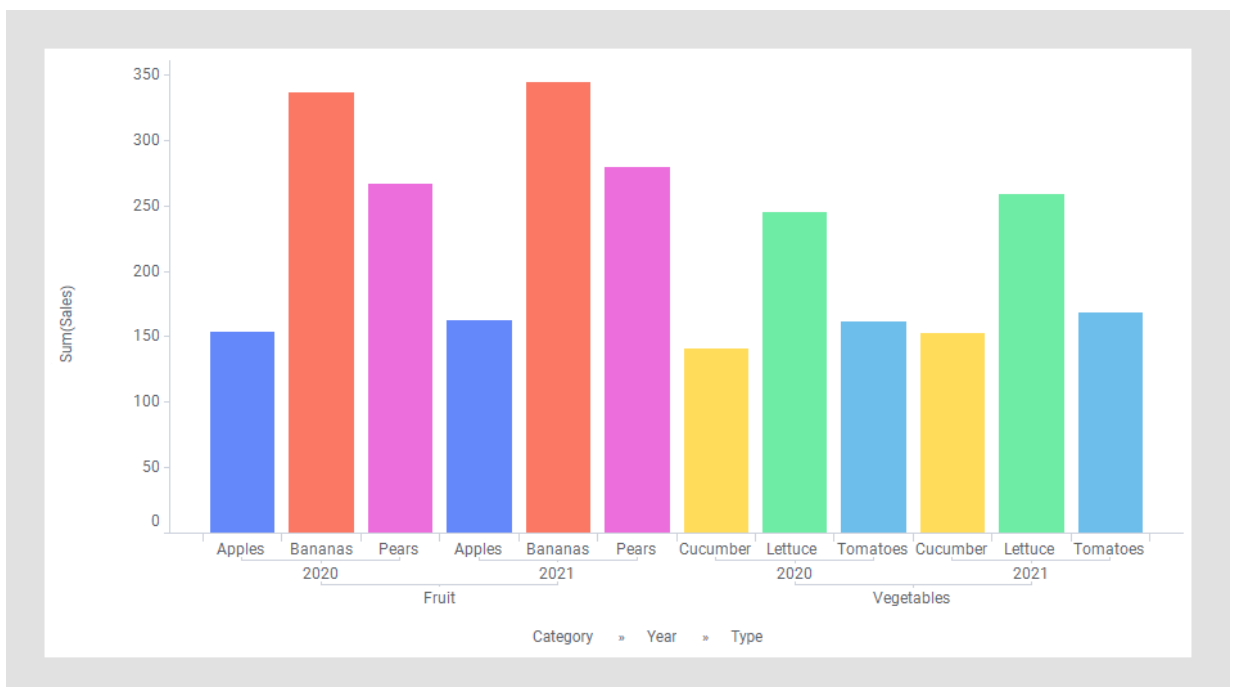
In the example below, three columns are selected on the category axis. The data becomes organized hierarchically with three different levels of detail. The bar chart automatically displays a bar for each combination in the hierarchy -- in this case the sales of each type of fruit and each type of vegetables per year.



You can change the hierarchy order, if you prefer to see the bars in another way. Drag the Category axis selector and drop it to the left of the Year axis selector.



This causes the bar chart to change the order of the bars. As you can see below, the bars concerning fruit sales are now placed next to each other. Note how the hierarchy displayed in the labels below the bar chart corresponds to the order of the axis selectors.



In Spotfire there are several ways to create and work with hierarchies.

- [Defining a hierarchy from a visualization axis](#)
- [Defining a hierarchy using the Add hierarchy dialog](#)
- [Defining a time hierarchy from a time-series column](#)
- [Additional operations with time hierarchies](#)

By adding more than one column to a visualization axis you can create a temporary hierarchy. You can edit this hierarchy any time by switching the order of the columns, as shown in the example above, or by adding or removing columns.

If you want to keep a certain hierarchy, that is, if you want the columns and their order to be fixed, and reuse it, you can [define the hierarchy](#). You can define a hierarchy either [from a visualization axis](#), from the [Add hierarchy dialog](#) or [from a time-series column](#).

A defined hierarchy is shown in the Data in analysis flyout as a column. The hierarchy can then be used in any visualization just like a column. It is also associated with a [hierarchy filter](#), which consists of an expandable tree view where categories at the different levels are represented by check boxes.

In addition, a hierarchy selected on an axis is supplemented with a slider, by which you can easily switch between the various hierarchy levels. The visualization will immediately adjust to the current level of detail.

Time data is by nature structured hierarchically. For more information about hierarchies with time-series data and how to work with them, see [Defining a hierarchy from a time-series column](#) and [Additional operations with time hierarchies](#).

## Defining hierarchies

When working with hierarchies, you can either create a temporary hierarchy or you can define one. You can edit a temporary hierarchy anytime by adding or removing columns, or by switching the order of the columns.

For an example of a temporary hierarchy, see [Hierarchies](#).

In a defined hierarchy, the columns and their relative positions are fixed. A defined hierarchy is shown in the Data in analysis flyout and behaves like a column. This means that it can be used in any visualization by adding it to an axis.

There are three major ways of defining a hierarchy:

- [From a visualization axis](#)
- [From the Add hierarchy dialog](#)
- [From a time-series column](#)

The created hierarchy is associated with a hierarchy filter which becomes available in the Data in analysis flyout and in the Filters panel. For more information, see [Using hierarchy filters](#).

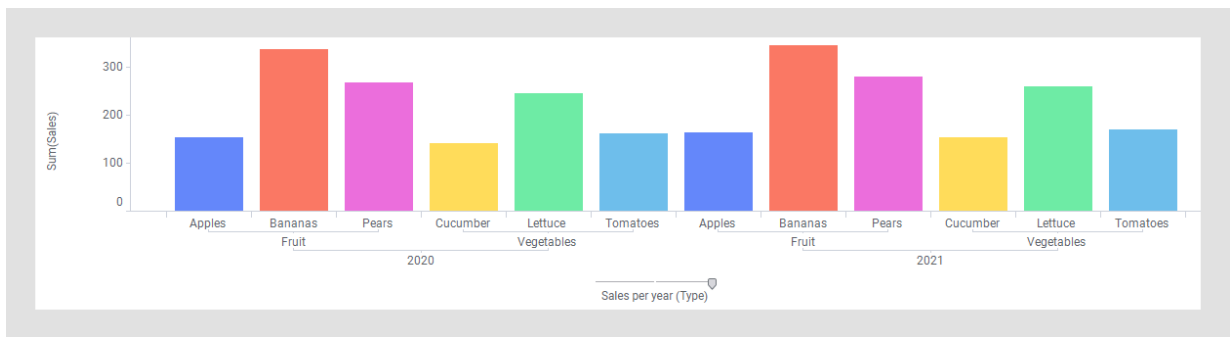
Adding a defined hierarchy to a visualization axis will display a hierarchy slider that makes it possible to easily switch between the levels in the hierarchy.

### Example

Suppose you have a defined hierarchy called Sales per year which contains Year/Category/Type. Add this hierarchy to the axis selector of the category axis of a bar chart as you would [select a column](#).

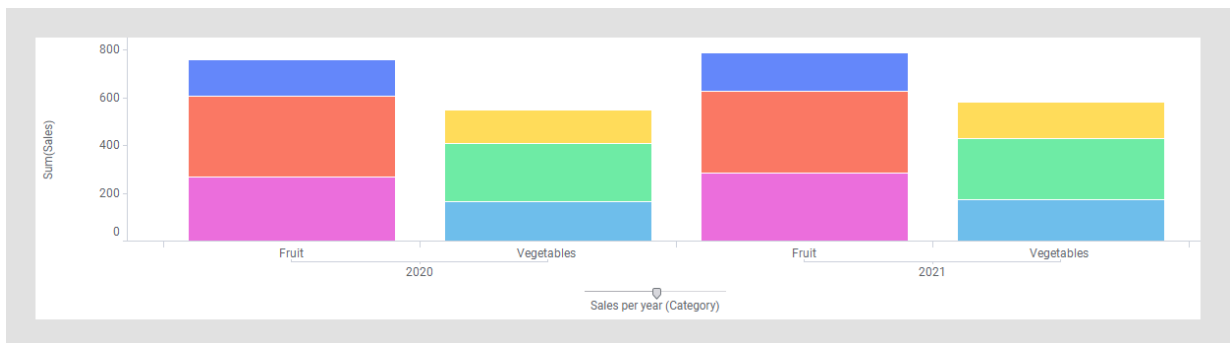


For more clarity, coloring has been applied.

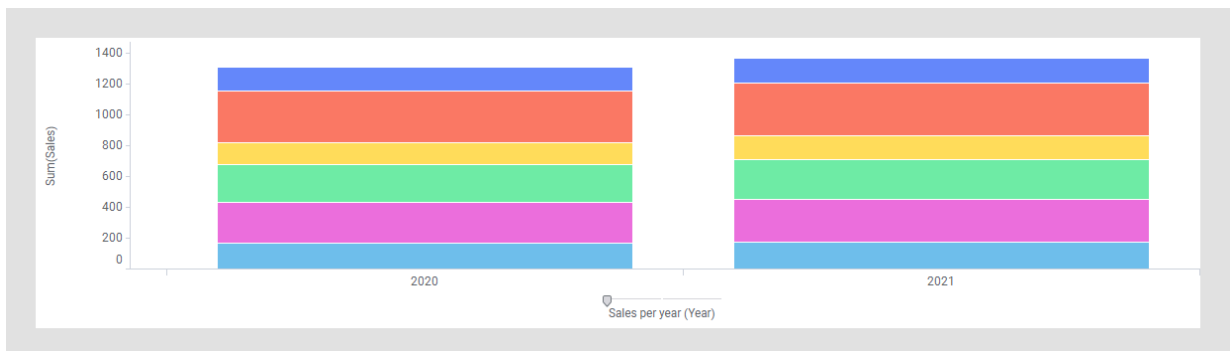


The labels underneath the bar chart state which bars represent the sales for each separate type, but they also show which types are part of which category, and which categories have been sold in a certain year.

There is only one axis selector, unlike in the example where [many columns are used on the axes](#) to build a hierarchy. Note that there is a slider on top of the axis selector. This hierarchy slider appears for all defined hierarchies and you can use it to change the level of detail of the visualization. If you drag the handle of the hierarchy slider one step to the left, this is what happens to the bar chart:



The bars are now automatically updated to show the total sales for each category per year, instead of for each type. Dragging the handle yet another step updates the bar chart to show the total sales for each year.



## Defining a hierarchy from a visualization axis

You can create a defined hierarchy directly from the columns on an axis in a visualization, for reuse in other visualizations or to obtain a hierarchy filter.

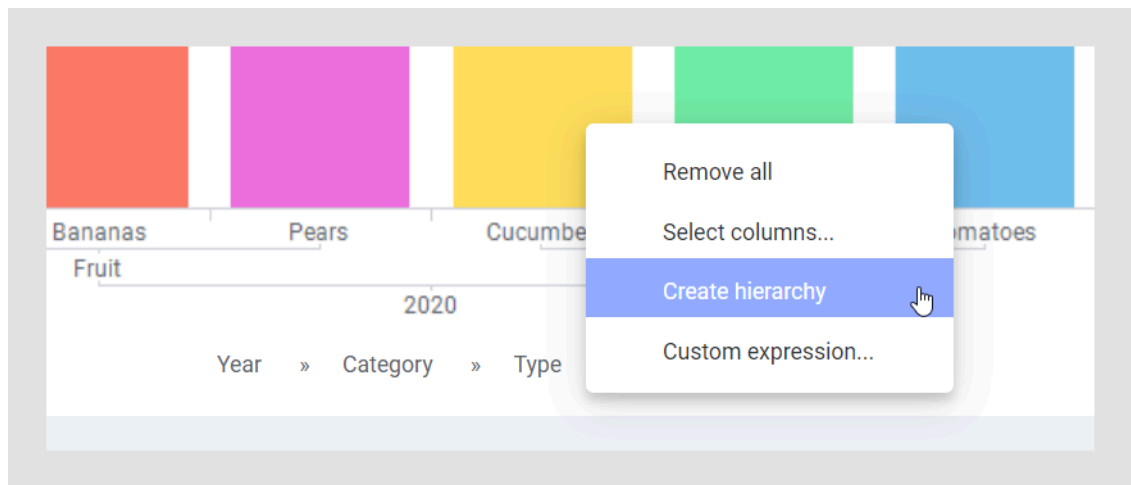
### Procedure

1. On the axis, select the columns to form the hierarchy and [place them in the wanted order](#).




If the number of columns added to an axis exceeds three, the name of each column will no longer be displayed. Instead the number of added columns is shown.

2. Right-click one of the column selectors, and select **Create hierarchy**.  
The separate column selectors are replaced by a single selector showing the hierarchy. The selector is supplemented with a slider, which allows you to switch between the levels of the hierarchy.




If you want, you can give the created hierarchy a name.

- a. Click **Data in analysis**  on the authoring bar to open the flyout.
- b. In the flyout, locate the created hierarchy, right-click it and select **Rename**.
- c. Enter a suitable name, and click **OK**.

## Defining a hierarchy using the Add hierarchy dialog

You can define a hierarchy for use in many visualizations or to obtain a hierarchy filter using the Add hierarchy dialog.

### Procedure

1. Select **Data > Add hierarchy** on the menu bar to open the Add hierarchy dialog.  
There are more ways to reach the Add hierarchy dialog:
  - From the [Data canvas](#): Click the plus sign on the final data table node, and select **Add hierarchy**.
  - From the [Data in analysis flyout](#): Hover with the mouse pointer over any column and click the Show filter button . Right-click the opened filter, and select **Add hierarchy**.
2. If more than one data table is available in the analysis, specify which **Data table** to work on.
3. Select the columns you want to include in your hierarchy from the Available columns list, and click **Add >** to show the columns in the Hierarchy list

4. Select a column in the Hierarchy list, and use the **Move up** and **Move down** buttons to position it correctly in the hierarchy.



The least granular category should be placed at the top of the list, for example, Continent should be placed above Country, and Country should be placed above City.

5. Enter a name for the new hierarchy in the **Hierarchy name** field.
6. Click **OK**.

### Defining a time hierarchy from a time-series column

Time data is by nature structured hierarchically. When specifying a column with dates or times in a column selector, you have the option to change the default linear series of dates to a hierarchical view of dates in the form of years, quarters, months, weeks, days, and so on.

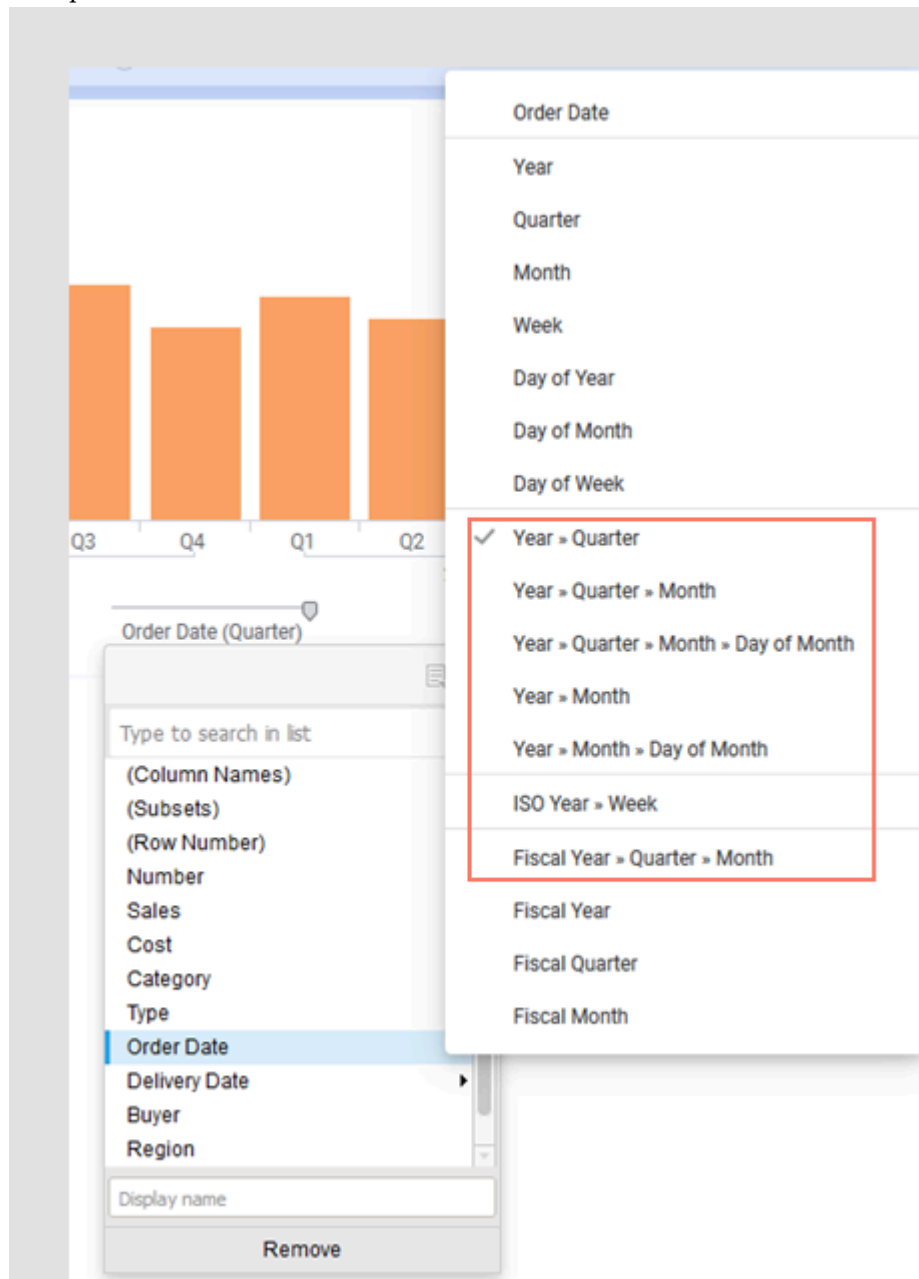
When you are working with time-series data (Date, Time, DateTime or TimeSpan columns), you can set up a hierarchical structure directly from the column selector.

#### Procedure

1. In the [column selector](#), click the arrow next to the column name. A popover listing all columns opens. Time-series columns have arrows indicating they can be expanded.



- Click the time-series column of your choice to expand a list of display options. The options highlighted in the image offer different time hierarchies, but you can also select a specific date or time part.



- Select the desired date/time part or hierarchy.

### Result

If a hierarchy was selected, the selected column is shown on the axis with a slider where you can specify which date or time part to inspect by adjusting the slider.

Due to the natural hierarchical form of time-series data, you can also perform [additional operations with time hierarchies](#), such as compensating for missing values.

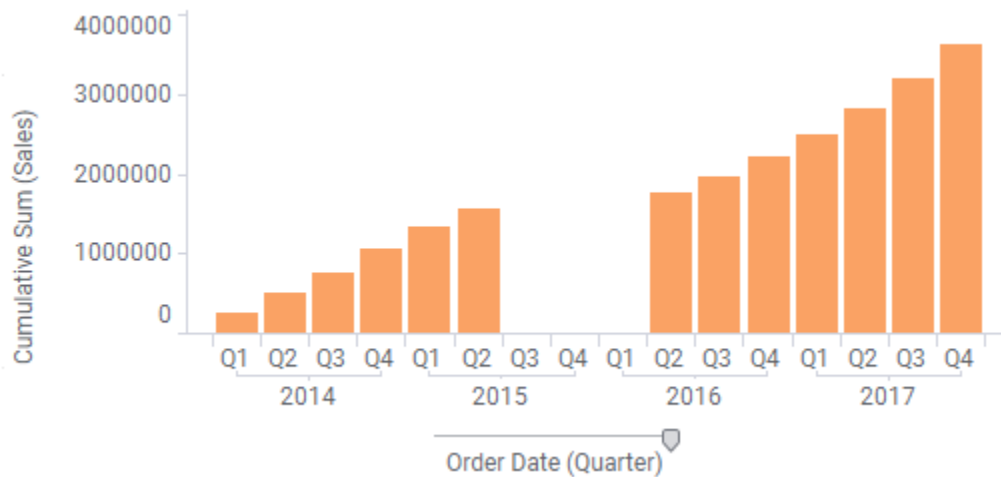
## Additional operations with time hierarchies

When working with time hierarchies, the expected time parts are known in a way that is usually not applicable for other types of hierarchies. Therefore, there are some additional operations available that can help you working with time data.

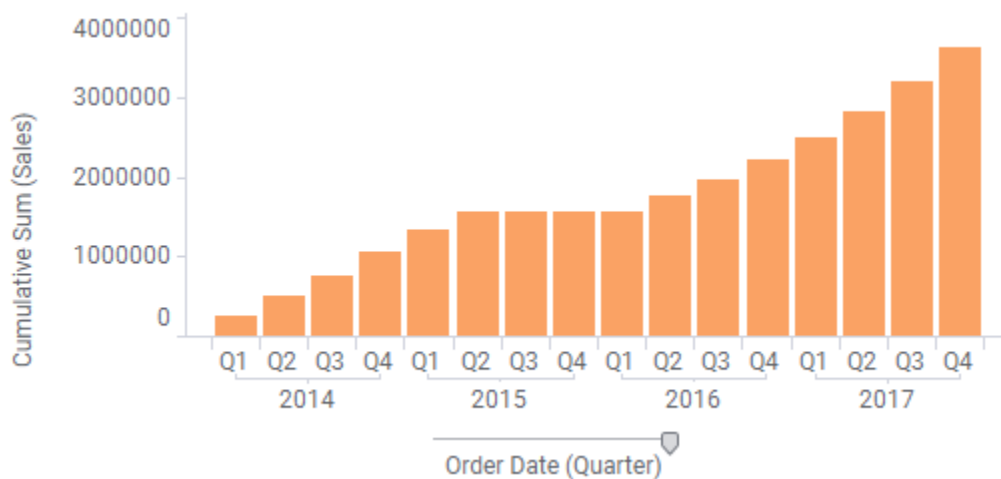
### Compensate for missing values

Sometimes you might work with data where some category values are missing. If the visualizations make use of a Date, Time or DateTime column and you like to present the data in an aggregated form, missing data can have strange effects on your calculations.

For example, below is a cumulative sum of sales for a few years, where data for three quarters are missing:



In this example, you might want the bars for the missing quarters to be of the same size as the last available bar rather than completely missing. To do this, you can use **Compensate for missing values** in the visualization properties. This functionality is available in the bar chart, the line chart, the combination chart (from the installed client only), the waterfall chart and the KPI chart. It introduces a gap-filling mechanism where empty values can be temporarily inserted for the missing rows. This way, the cumulative sum can instead show the same value as the last available bar for the bars with missing data:



In the web client, go to the visualization properties, and in a bar chart or waterfall chart, click **Axes**, and under **Category axis** select **Compensate for missing values**. In the line chart, the functionality is found under **Axes > X-axis**, and in the KPI chart under **Values > Time** (for a specific KPI).



You can use search in the web client visualization properties to find specific functionality, such as "Compensate for missing values".

In the installed client, you reach **Compensate for missing values** via the **Appearance** page.



Compensate for missing values only works with true Date, Time, or DateTime columns. The check box will have no effect on a hierarchy made by putting together string values from different columns (such as Year, Month, and Day).

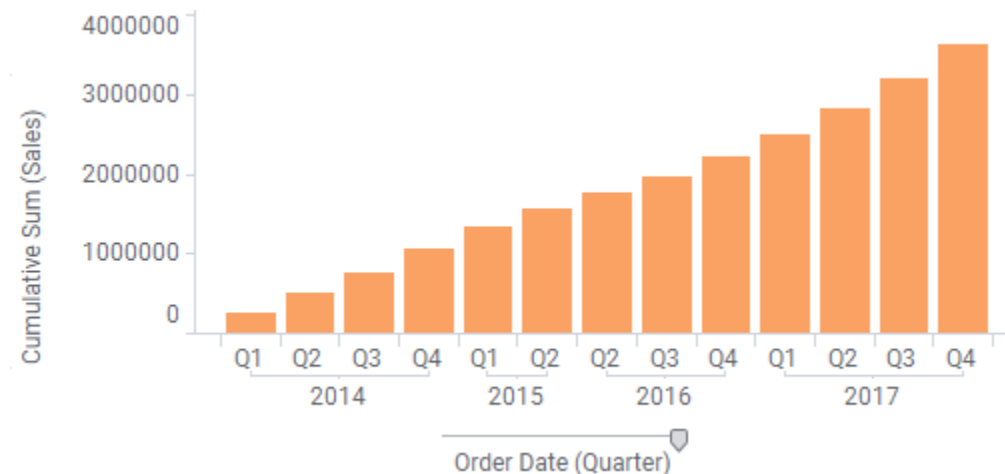
Data values in general can be valid values (non-null and non-error), invalid values (errors, such as a value of a different data type than the rest), or empty values (where the row identifier is there, but there is no value in the value column). There might also be completely missing categories. For most kinds of data, this can be difficult to handle, but time-series can be filled out with the missing data points, because it is always possible to calculate which time points should have been available. Spotfire can temporarily add the missing rows and fill them with empty values so they can be used in calculations.

When **Compensate for missing values** is used, the max and min values in the time column are determined. Then all missing steps, depending on the time hierarchy level, are added.

Multiple date/time parts on different axes can be combined in a visualization as long as the same date/time column is used.

### Categories (installed client only)

When categorical time values are used in a visualization, and you are working in the installed client, you can control what to show using the **Categories** setting on the axis. You can select the **Show Filtered Values** setting to completely hide those categories where no data is available (either due to filtering or missing data):



In the first image in this topic, there is an example of a visualization with missing values where no compensation for missing values has been made, and the **Show Filtered Range** setting has been selected. This option only hides empty categories on each side of the range, but not in the middle. There is also an option to **Show All Values** where the currently filtered-out values on the sides also remain visible.

In the installed client, you can change the setting from the Advanced Settings dialog (reached from the visualization properties for the axis of interest) or by right-clicking on the visualization axis and selecting **Categories** and one of the options from the pop-up menu.

To work with continuous time axes, use zoom sliders instead.

## Mixed hierarchies

Mixed hierarchies with both date and/or time parts (from the same DateTime column) and other categories are supported by treating the categories as part of the group by hierarchy. You can specify the combinations to show using a custom expression. In the installed client, you can also specify which combinations to show in the Advanced Settings dialog.

## Max number of added rows

If the range of the date/time column is significantly larger than the step size, a huge number of rows might be needed when compensating for missing values. For example, this will be the case when the step size is millisecond and you have dates spanning several years. For performance reasons a property limit is defined, determining the max number of rows that can be inserted. In the installed client, the **MaxMissingTimeParts** property can be changed under **File > Document properties > Properties**. An administrator can also change this preference by going to **Tools > Administration manager > Preferences tab**, and then clicking on **DataOptimizationPreferences** under **DataOptimization** and editing the **MaxMissingTimeParts** preference.

Note that the limit for max number of rows to add is based on the size of the span and not on the actual missing values. This means that if the preference value is set too low, you might encounter this limit even though not that many time parts were missing in the current setup. If the message "Could not compensate for missing values. The document property value for MaxMissingTimeParts has been exceeded." is encountered, you might want to increase the property value, but the solution can also be to clear the **Compensate for missing values** check box or simply to try to reduce the granularity on the time series axis by using a hierarchy slider (if one is available).

## Marking temporarily added rows

When a visualization item based on temporarily added rows is marked it will look like a regular marking in the visualization. However, because no real rows are marked, this marking will not be propagated to any other visualizations nor to the Details-on-Demand.

## BinByDateTime

A time hierarchy is actually built using the [BinByDateTime](#) function. This function uses three arguments:

1. the Date, Time or DateTime column
2. the hierarchy definition as a string of the date parts, separated by dots (e.g., "Year.Month")
3. the pruning level (meaning which level to start the hierarchy slider position on)

For example, `BinByDateTime([Column], "Year.Quarter.Month.Day", 2)`

For Date or DateTime columns you have the option to show hierarchies based on the ISO 8601 standard by using the ISOYear and ISOWeek date parts, rather than the regular year and week date parts. According to ISO 8601, Monday is always the first day of the week, and the first week of a year is the week that contains 4 days or more from the new year. An ISO week-numbering year contains 52 or 53 full weeks.

Use an ISO week hierarchy by defining a BinByDateTime expression similar to `BinByDateTime([Column], "ISOYear.ISOWeek", 1)`, or select the **ISO Year >> Week** hierarchy directly from an axis, as seen on [Defining a time hierarchy from a time-series column](#) on page 508.



The `Year`, `Week`, and `YearAndWeek` methods depend on regional settings, and might produce strange results for values crossing a year boundary (for example, you might see week 53 for some days when it should have been week 1). To get predictable results, use `BinByDateTime([Column], "ISOYear.ISOWeek", 1)` or the separate `ISOYear/ISOWeek` methods instead.

## BinByTimeSpan

Similarly, a time span hierarchy is built using the [BinByTimeSpan](#) function. Binned time spans can be useful, for example, when viewing groups of intermediate times in a sporting context or when analyzing log data. This function uses three arguments:

1. the `TimeSpan` column
2. the hierarchy definition as a string of the time span parts, separated by dots (e.g., "Hours.Minutes")
3. the pruning level (meaning which level to start the hierarchy slider position on)

For example, `BinByTimeSpan([Column], "Days.Hours.Minutes.Seconds", 2)`

## Editing a defined hierarchy

When you have defined a hierarchy using **Create hierarchy** on the visualization axis, or, using the Add hierarchy dialog, you can edit the hierarchy from the data canvas.


### Prerequisites

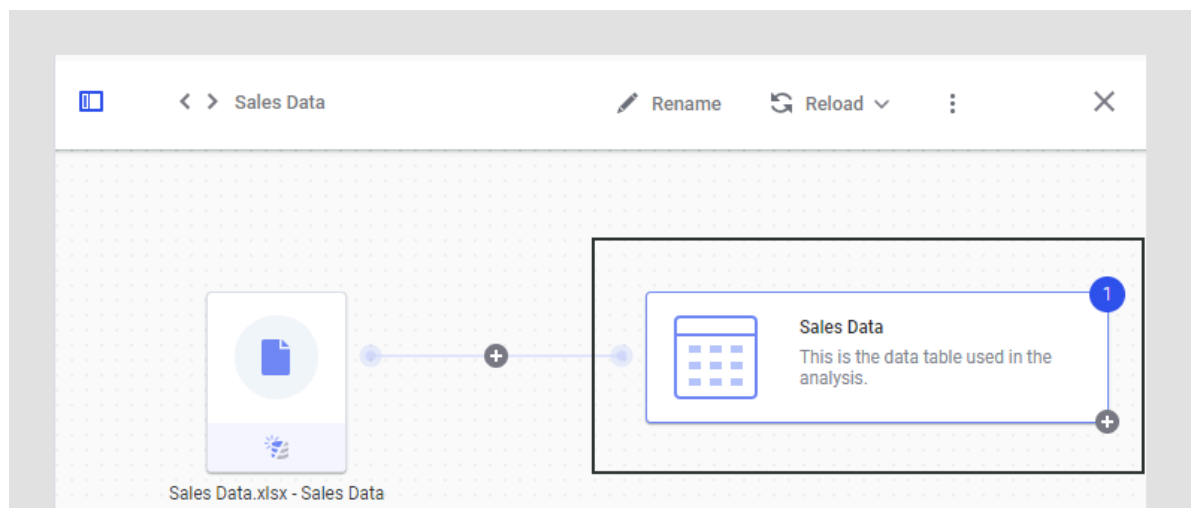
A defined hierarchy has been created as described in [Defining a hierarchy from a visualization axis](#) on page 507 or [Defining a hierarchy using the Add hierarchy dialog](#) on page 507.



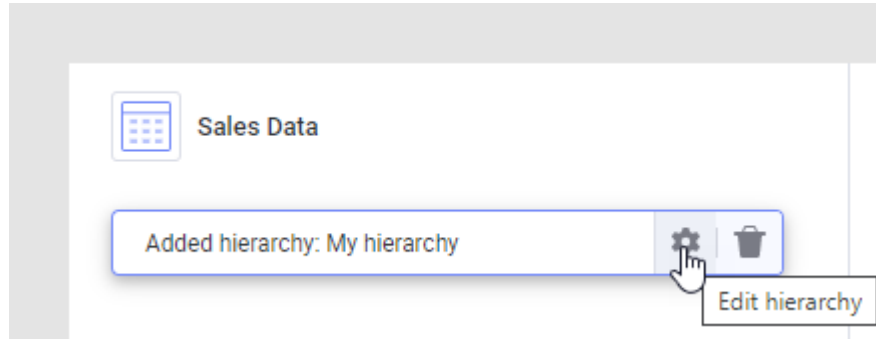
Temporary hierarchies on axes can be [rearranged by dragging columns](#) on the axis, or by removing or adding columns. [Time-series](#) hierarchies can be removed from the axis and re-added with another structure, or, the underlying binned custom expression can be edited by right-clicking on the axis and selecting **Custom expression**.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Navigate to the data table of interest, and, in the data canvas, click on the final node (to the far right).



3. In the lower left part of the data canvas, click **Edit hierarchy**.



4. In the Edit hierarchy dialog, make the desired changes and click **OK**.

### Result

The hierarchy is updated.

## Aggregating data

Visualizing data involves presenting aggregated values of the data you have loaded. Examples of aggregated values are sums, averages, counts of occurrences, or results from various statistical calculations. That is, they represent collections of data as single values. In most cases, the data to aggregate is numerical, but also non-numerical data can be aggregated.



Some visualizations, like the table visualization, do not support aggregation of values, and sometimes values can be aggregated only on certain axes.

The resulting values of an aggregation reflect the current data. That is, when you filter the data, a recalculation takes place instantly, and the visualization updates.

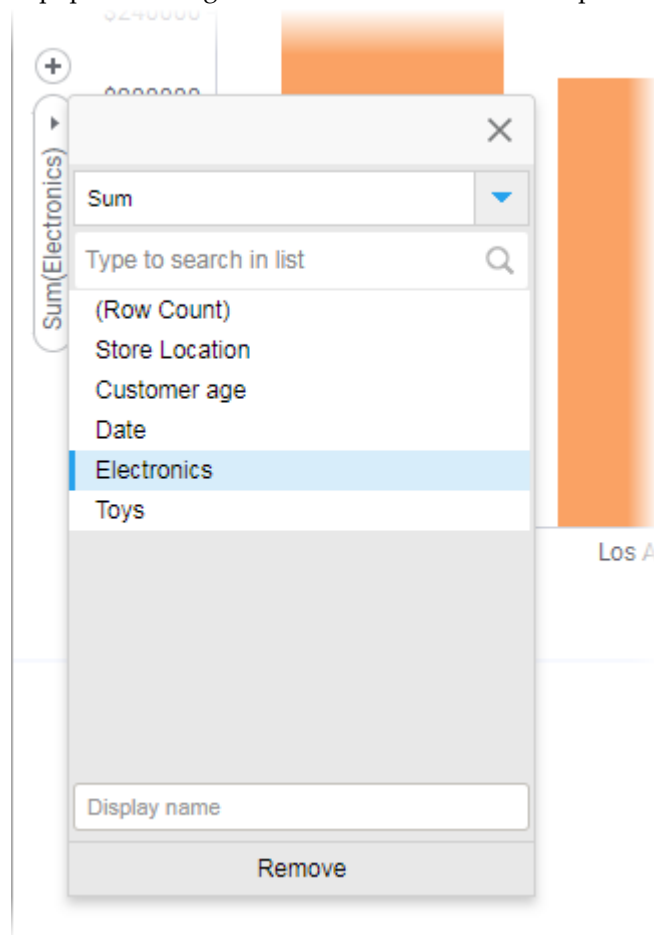
You specify which aggregation to display on the axis' column selector .

### Prerequisites

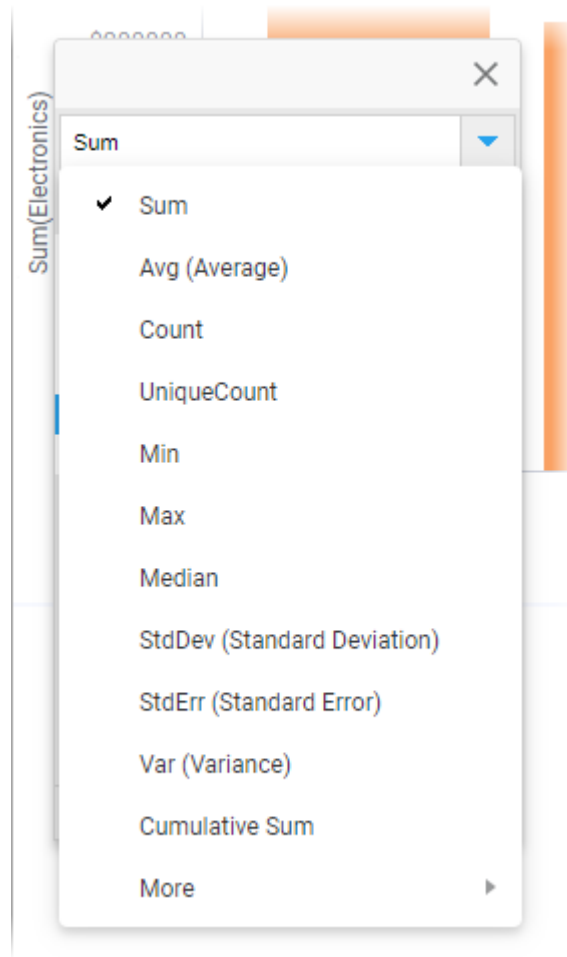
The data column of interest has been specified in the axis' column selector.

## Procedure

1. Click the arrow next to the column name in the column selector.  
A popover listing all columns in the data table opens. The selected column is colored.



2. Click the blue arrow.  
A list of available aggregations opens.



You can click the arrow at the bottom to display more aggregations.

3. Select the aggregation to display on the axis.



In the field at the bottom of the first popover, you can [specify your own display name for the axis](#).

4. Click anywhere outside the popover to close it.

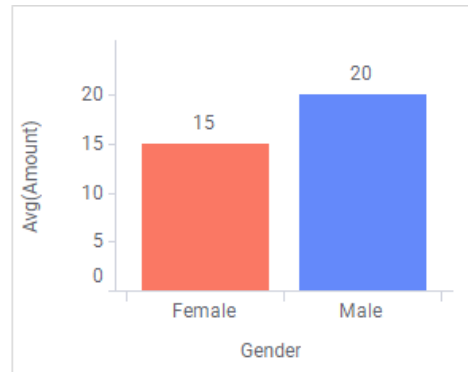
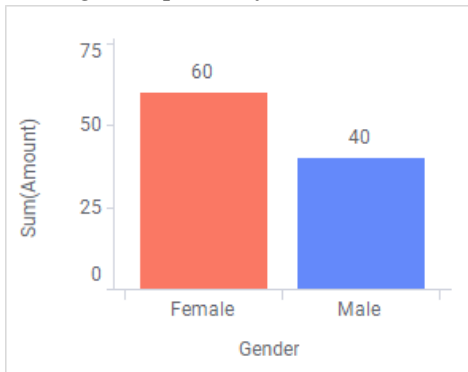


## Examples of aggregations

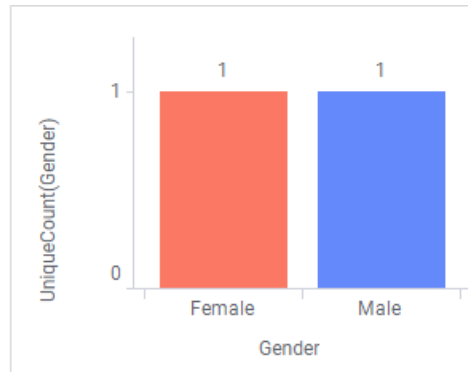
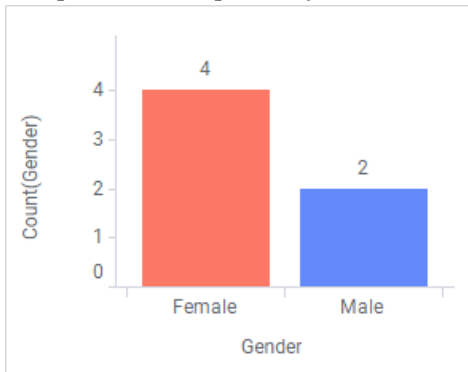
To illustrate some aggregations, this simple data table is used.

Customer ID	Gender	Amount
C011	Female	10
C012	Female	15
C013	Male	35
C014	Female	15
C015	Female	20
C016	Male	5

First, data in the numerical column 'Amount' is aggregated using Sum and Avg (average), respectively.



Second, data in the non-numerical column 'Gender' is aggregated using Count and UniqueCount, respectively.



## Adjusting scales on axes

By default, there is only one scale on an axis, and all measures are shown on this scale. If the measures are of completely different magnitudes, or use different units, you might benefit from using other options, for example, adding more than one scale on an axis to visualize the measures.

To give an example, imagine a bar chart containing two bars, each of the bars representing a certain column. One bar represents the total precipitation during a month, and another bar the average temperature. These two measures probably differ a lot, and might have need for two scales on the value axis.

You might encounter some of the following scale options in the various visualization types.

- **Single scale**

The **Single scale** option means that the scale is common to all the measures on the axis. It is the default scale setting.

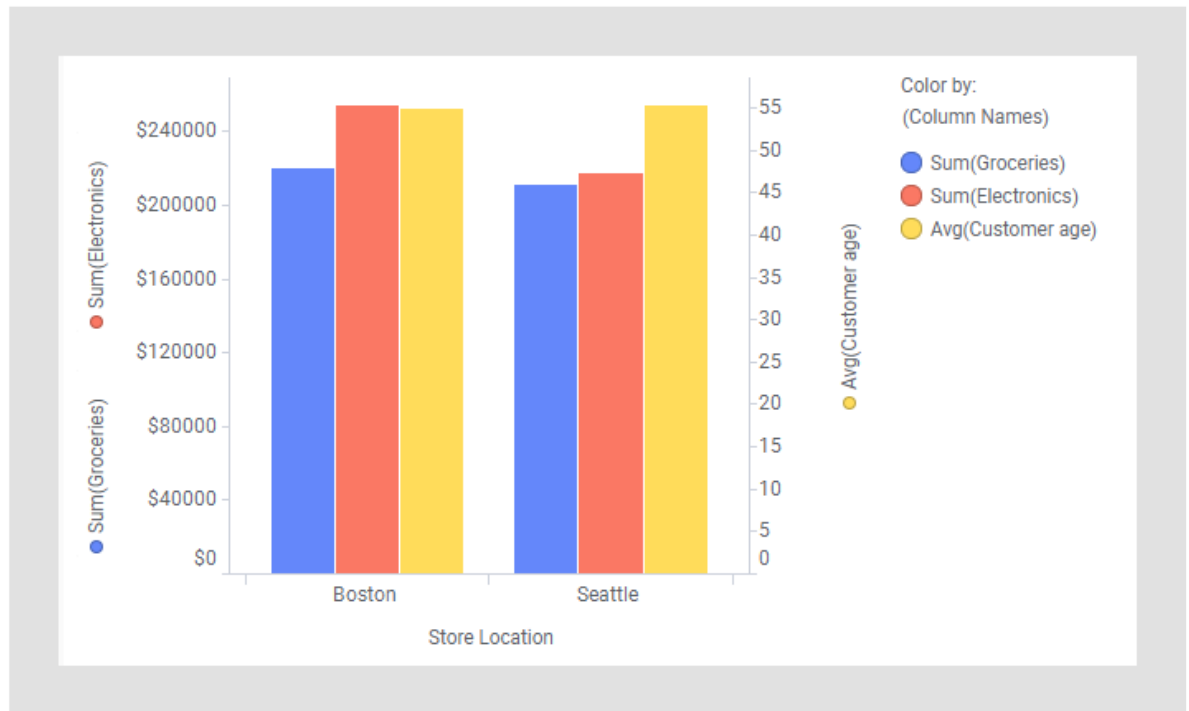
For example, on the value axis below, there is only one scale, shared by both the measures, Sum(Electronics) and Sum(Toys).



- **Dual scales**

When you use the **Dual scales** option, there is one scale to the left and one scale to the right on the Value axis/Y-axis. Each of the scales can be shared by more than one measure.

In the example below, the value axis has two scales, one to the left and one to the right. Two columns share the left scale, and to the right, another scale is used for the third column, because the unit is different and the values are of totally different magnitudes.



This option is available in bar charts, line charts, combination charts, and scatter plots.

- **One scale per color**

Another option to display more than one scale is **One scale per color**. If you use this option, you get one scale for each color that is defined on the color axis. This means that no scales are shared.

An example of a value axis with three scales is shown below. The scales are differentiated by their colored scale labels.

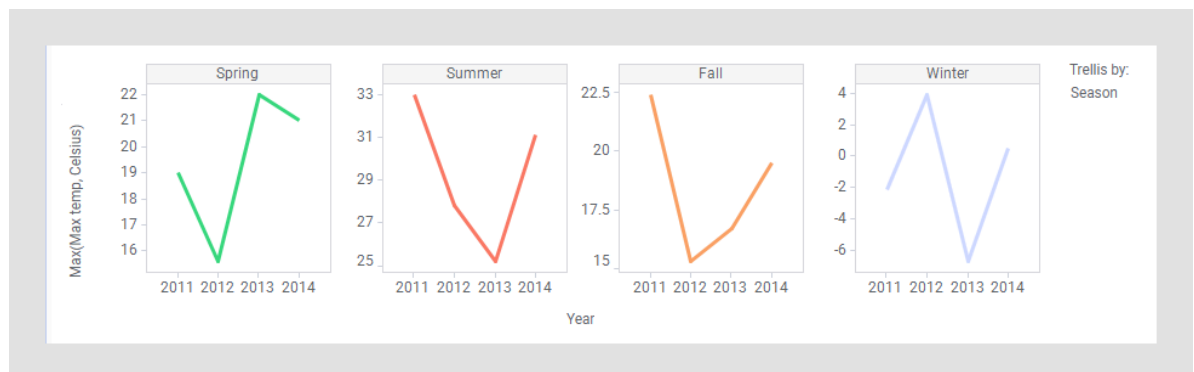


This option is available in bar charts, line charts, combination charts, and scatter plots.

- **One scale per trellis panel**

For [trellised visualizations](#), it is possible to display one scale per trellis panel.

For example, the line chart below shows the maximum temperature measured per year at a certain location, and it is trellised per season. The scales in the different trellis panels are adapted to the seasonal temperatures.

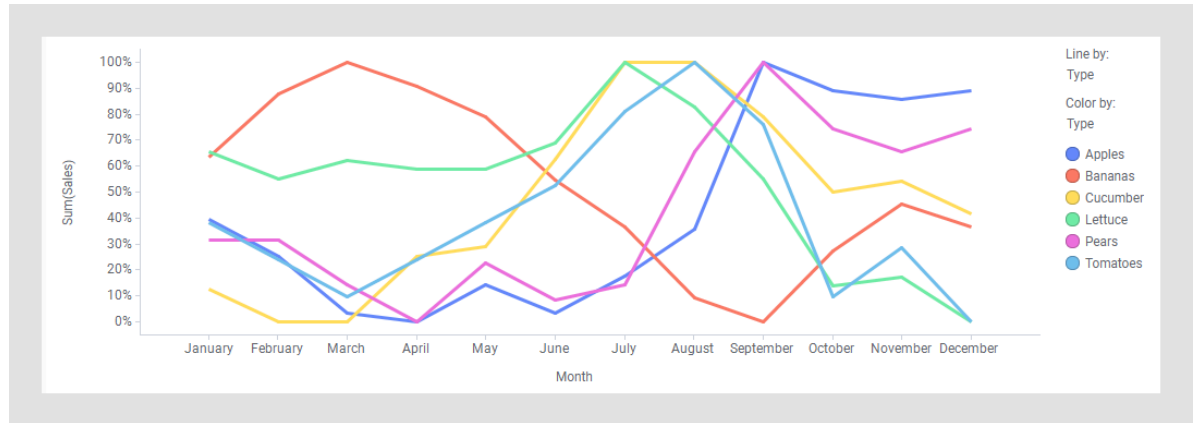


This option is available in bar charts, line charts, combination charts, waterfall charts, and scatter plots.

- **One scale per line**

Moreover, in a line chart you have the option to display the lines normalized, so that all lines are drawn on a scale with the maximum value for each line set to 100%, and the minimum value for each line set to 0%.

In the example below, the normalized lines are used for visualizing sales patterns.



### Procedure

1. Right-click the visualization, and select **Properties**.  
The Properties pop-over is displayed.
2. Click **Axes**.  
The Axes section is displayed.
3. Beneath **Value axis** (applicable to bar charts) or **Y-axis**, select the scale option of interest.

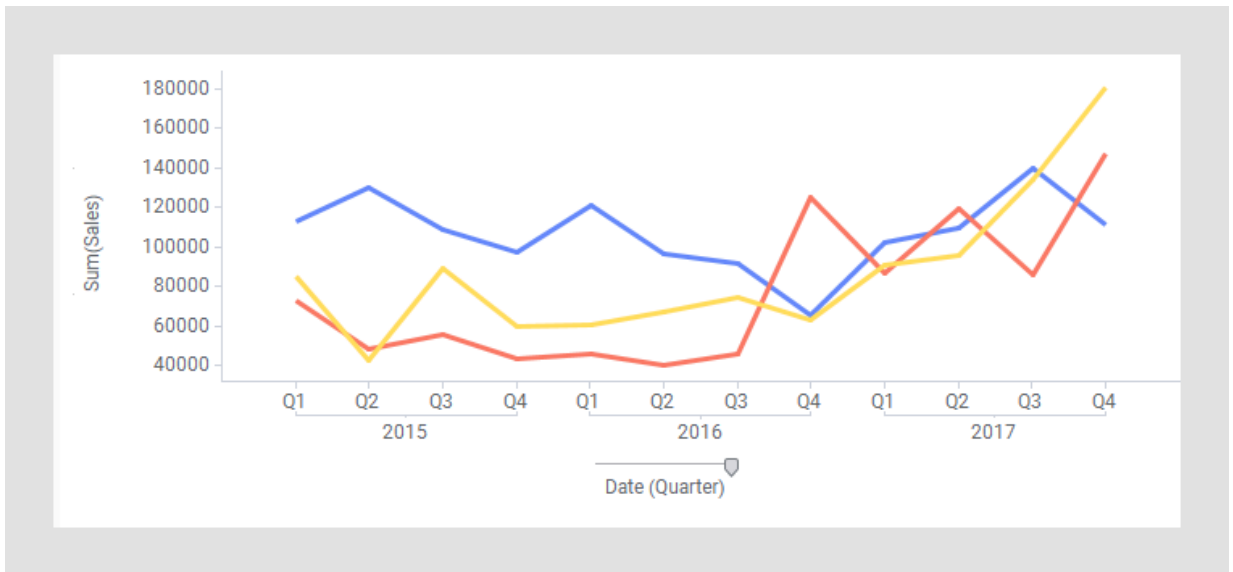


Below the scale options, you can specify the formatting of the scales.

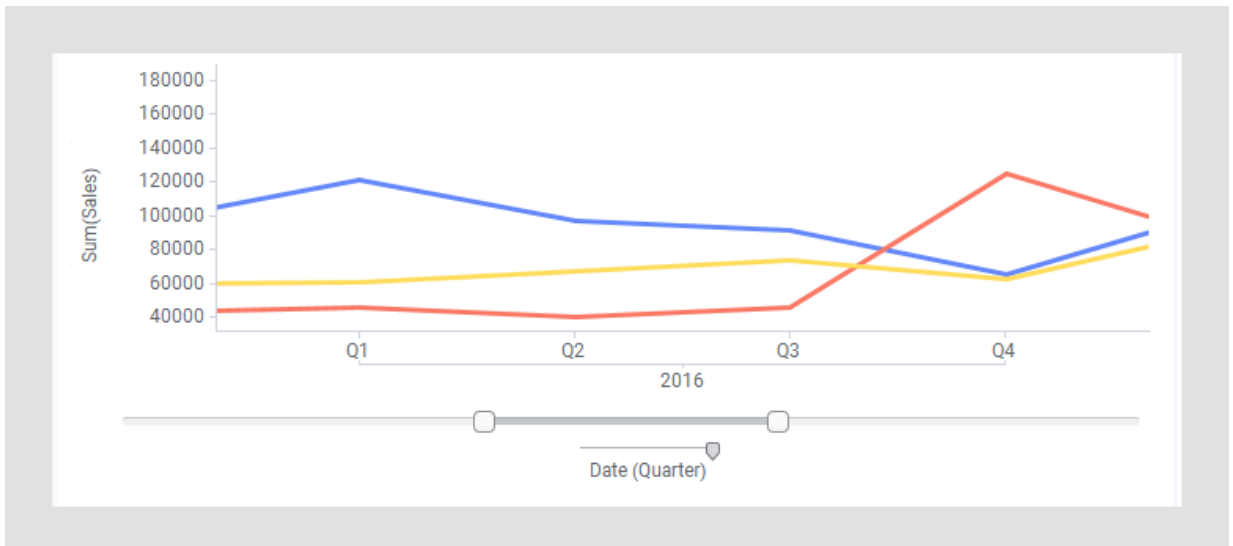
## Adding a zoom slider

Zoom sliders are used to get a closer look at details in your visualization.

The line charts below illustrate the use of a zoom slider. The line chart at the top shows sales from Q1 2015 to Q4 2016.



By adding a zoom slider to the line chart, and then adjusting it by dragging its handles, you can look closer at, for example, the quarters in 2016.



## Procedure

1. Right-click the visualization, and select **Properties** in the opened menu.  
The Properties pop-over is shown.
2. Click **Axes**.  
The Axes section is shown.
3. Beneath the axis you want to supplement with a zoom slider, select **Show zoom slider**.  
A slider is added to the axis.
4. Zoom in to details by dragging the handles of the zoom slider.



You can grab the gray part of the zoom slider and drag the slider to the left or right to pan horizontally.

## Changing display name for an axis

You can specify your own display name for an axis.

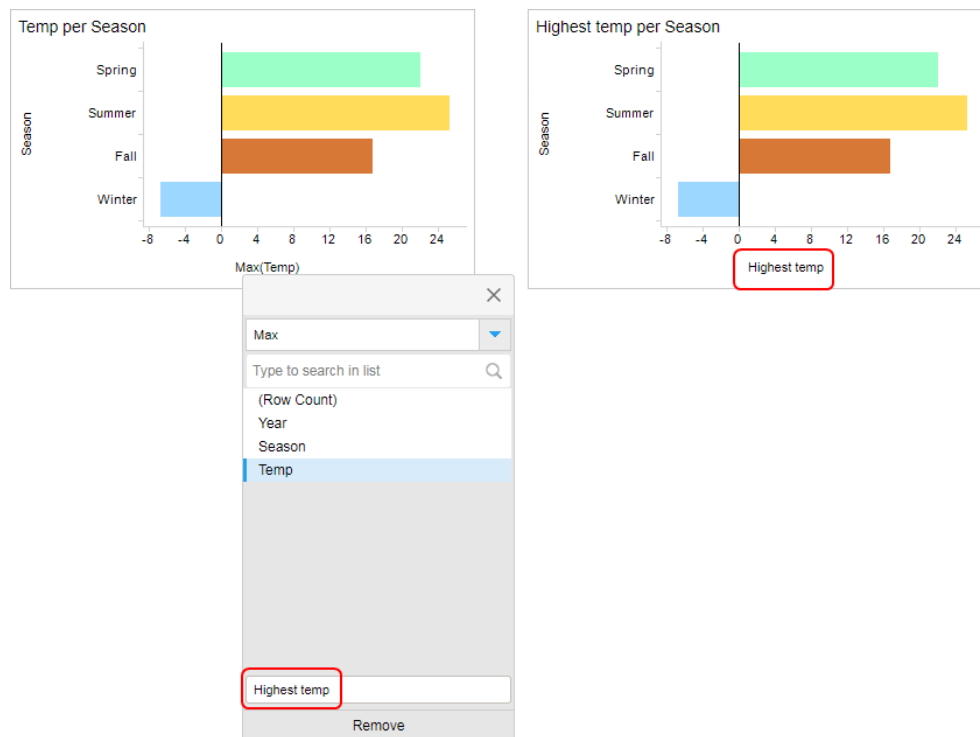
The display name can be changed for any axis, for example, the X-axis, the Y-axis, or the Color axis.

### Procedure

1. In the column selector on the axis, click the arrow next to the column name.  
A popover opens that shows which column and aggregation are selected on the axis.
2. In the field at the bottom of the popover, enter the wanted display name and press Enter.  
The new display name becomes visible on the axis.
3. Click outside the popover to close it.

### Example

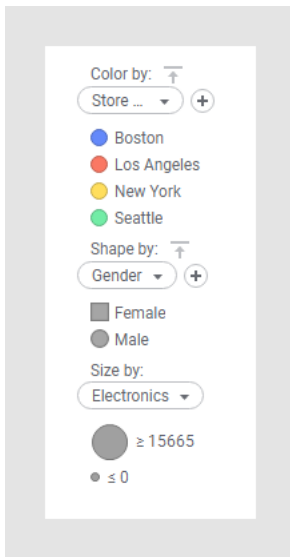
In the bar chart below, the display name on the Value axis is changed from Max(Temp) to Highest temp. The visualization title adjusts automatically.



## Showing/hiding the legend

The legend provides information that helps the viewer identify what is represented in a visualization. You can decide whether or not the legend should be visible.

An example of a legend is shown below:



Show the legend when the information within it is necessary for interpreting the visualization. Hide the legend when you must use as much as possible of the screen estate for the visualizations themselves.



You can decide [what information to include](#), and the position of the legend within the visualization.

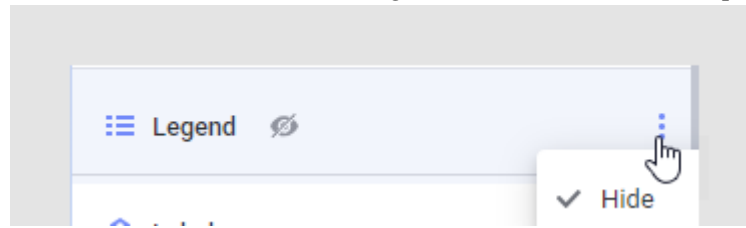
### Procedure

1. Place the cursor on the visualization.  
Icons are shown in the upper right corner of the visualization.



2. Click to toggle between showing and hiding the legend.

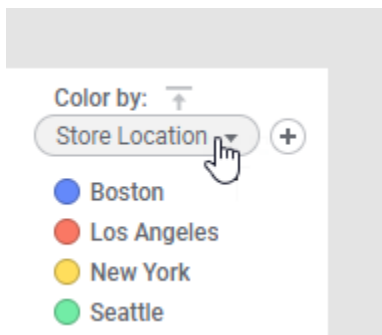
You can also show or hide the legend from the visualization properties.



### Result

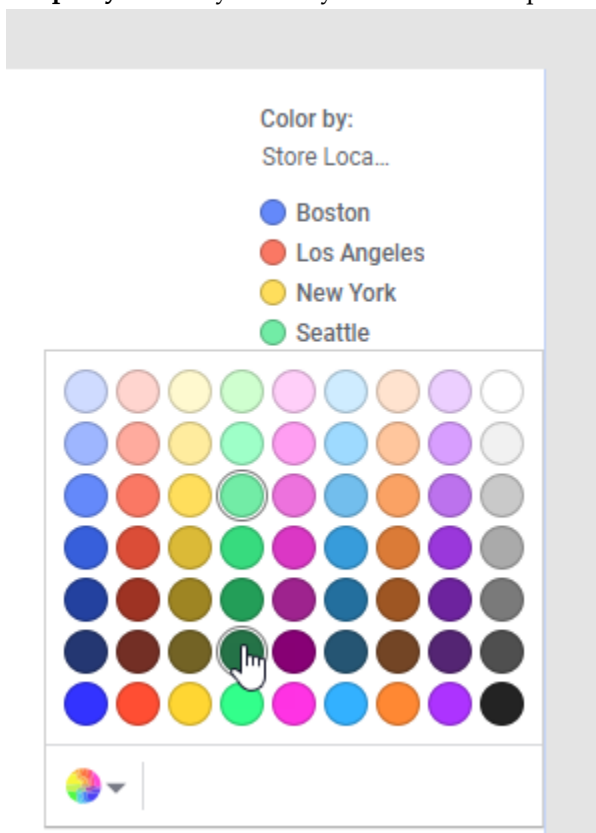
Which information is available in the legend by default varies with the visualization type. For example, the legend of a pie chart will show which column determines the size of the pie sectors, and the legend of a scatter plot will show if a column has been used to define the marker size or marker shape, and so on. You can edit all such properties by clicking the respective column selectors in the legend.





When multiple data tables are available in the analysis, the legend for all new visualizations will automatically show a data table selector where you can switch data table.

You can also change the colors used in the visualization directly from the legend. Click the color circle in the legend to open a color palette where you can select another color. Clicking a shape icon under **Shape by** similarly allows you to switch shape.




## Specifying the legend appearance

The legend provides information that helps the viewer identify what is represented in a visualization. You can specify the position of the legend within the visualization and what items it should include.

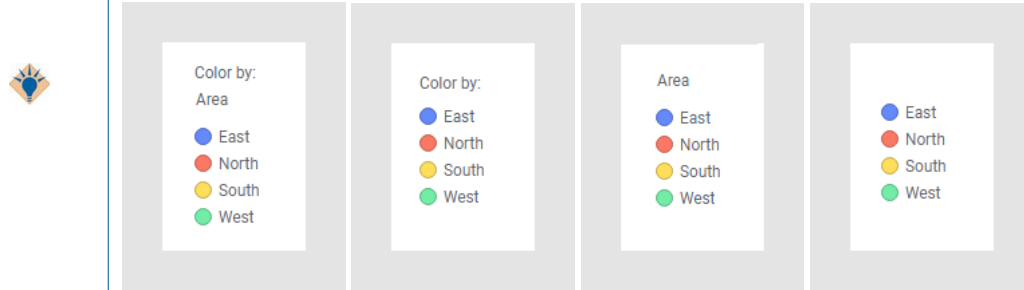
### Procedure



1. In the visualization properties, locate the **Legend** section.
2. Beneath **Position**, select **Right** to show the legend at the upper right of the visualization, or **Left** to show it at the upper left of the visualization.

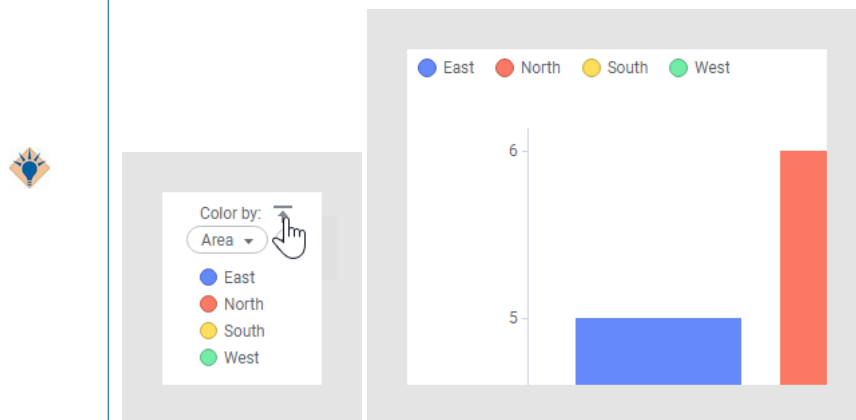
- Optional: Beneath **Max number of values per legend item** (you might need to click  first) enter a number to specify how many values you want to show per legend item, or use the arrows to increase or decrease that number.
- Beneath **Show the following legend items**, select the items you want to show, and remove items that should not be included in the legend.

Some legend items are visible only when they are actually used.

For some items, you can specify in detail what to show in the legend. Click an item in the **Legend** section, and select or clear the **Show title** and **Show axis selector** check boxes to get different display options. Examples are shown below:



By selecting and clearing the **Show values horizontally** check box, you can for some items decide to show their values horizontally at the top or the bottom of the visualization or to keep them in the vertical legend. To show the item values horizontally, you can also click this icon  next to the item title. To move the values back into the vertical legend, click  again. This is possible, for example, for the items **Color by** or **Shape by**.

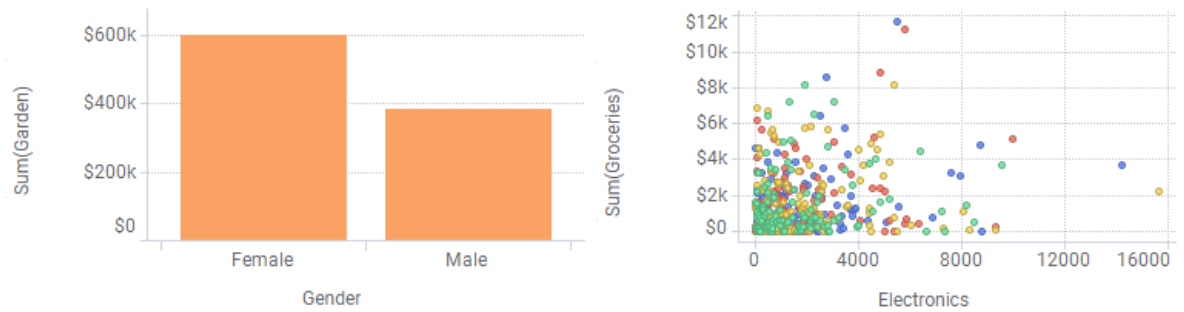


With horizontal legend items, you can also specify whether to place the items at the **Bottom** of the visualization, and choose the position to be **Left**, **Right** or **Center**. These settings are reached from the options menu to the right of the items.

## Showing gridlines

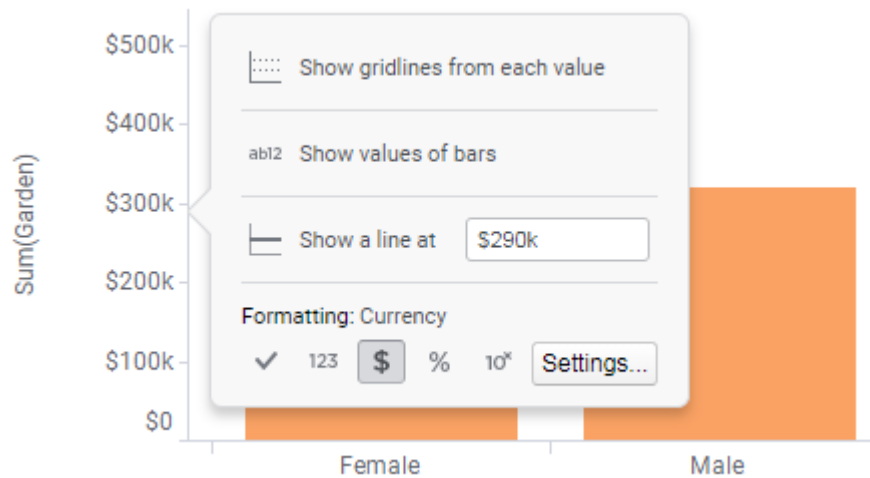
You can add gridlines to the visualization background to make it easier to get a reading of the value for an item. Gridlines are dotted lines that are drawn horizontally or vertically from the tick marks on numerical axes.

The visualizations are by default drawn without gridlines. Gridlines can be added on the value and category axes in bar charts, and the X- and Y- axes in line charts, combination charts, and scatter plots provided that the axes represent numerical values. Examples of visualizations with gridlines are shown below.



## Procedure

1. Place the cursor on the numerical axis.  
The axis is highlighted.
2. Click the highlighted area to access a pop-up menu.



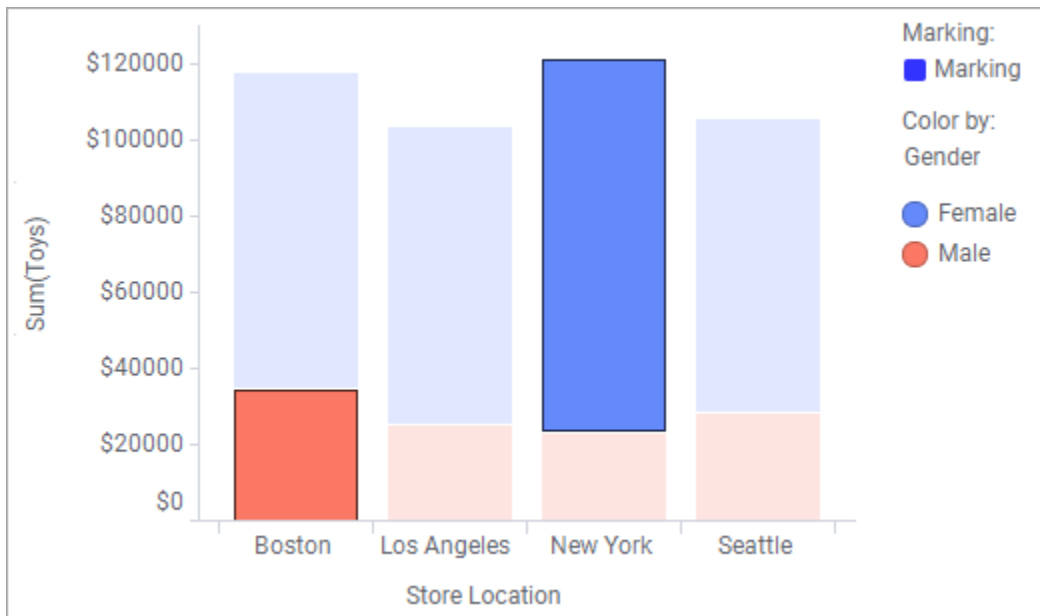
3. Select **Show gridlines from each value**.



If you want to remove the gridlines, repeat the steps above.

## Marking

The action of marking visualization items means selecting parts of the data that you find of certain interest to examine in more detail, or to work at in various ways. To gather the marked items, and to recognize them as marked, they are associated with a certain marking.



You mark items in different ways, for example by clicking them. Example of items are bar segments, markers, lines, and pie sectors. Items that have been marked are easily distinguished: the unmarked items' colors fade out, but the marked items keep their original colors and get a dark outline as shown above.



You can also choose to show marked items in a visualization with the defined marking color by selecting **Use separate color for marked items** in the visualization properties.

When you mark items in a visualization, you in fact mark the corresponding rows in the data table. To identify and represent this entire set of marked rows, a special concept is used, marking. A marking is assigned two attributes, a name and a color, and can be viewed in the legend, see the image above.

Because rows in the data table become marked, every visualization in your analysis that uses the data table and the same marking is affected. If items in other visualizations include any of the marked data rows, they become marked too.

How to mark and unmark items is described in

- [Marking items](#)
- [Marking items using the legend](#)
- [Unmarking items](#)

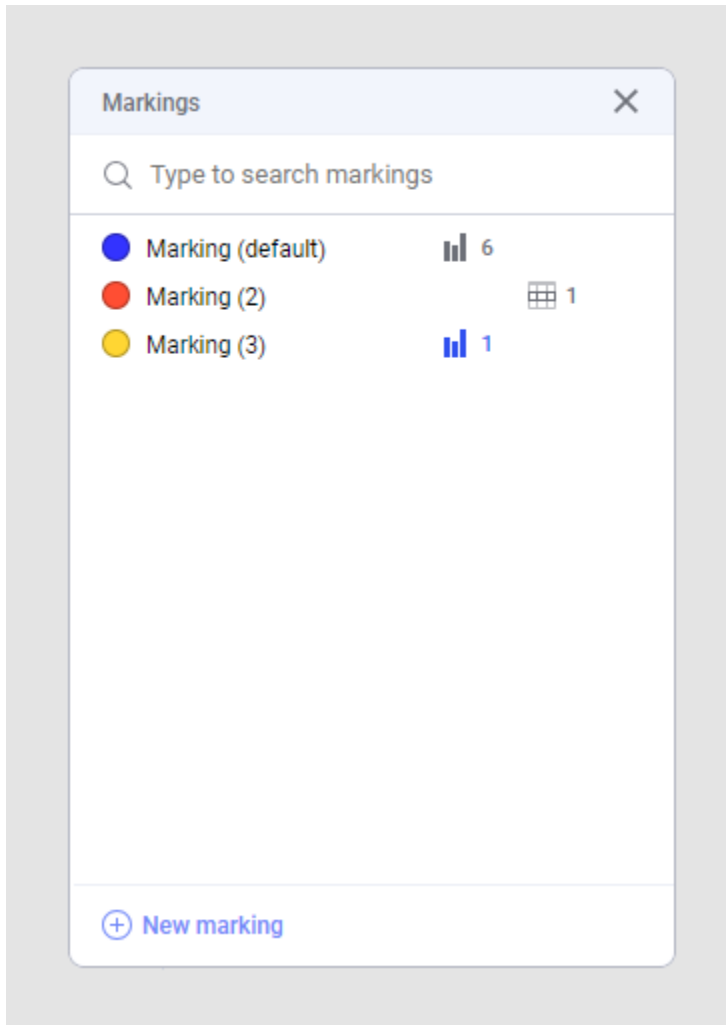
Marking items serves many purposes. For example, it can be used to drill down into the data or to group data in your analysis. Here you find different use cases:

- [Exploring data across visualizations](#)
- [Displaying item details](#)
- [Creating details visualizations](#)
- [Grouping categories](#)

You can add more than one marking to the analysis, if desired. See [Adding a new marking](#) on page 538 for more information. If there are more than one marking available, the default marking is used

when creating new visualizations, but you can change which marking to use in a specific visualization from the visualization properties.

In the Markings panel (**View > Markings**), you can see on which page, or in which visualization, a certain marking is used.



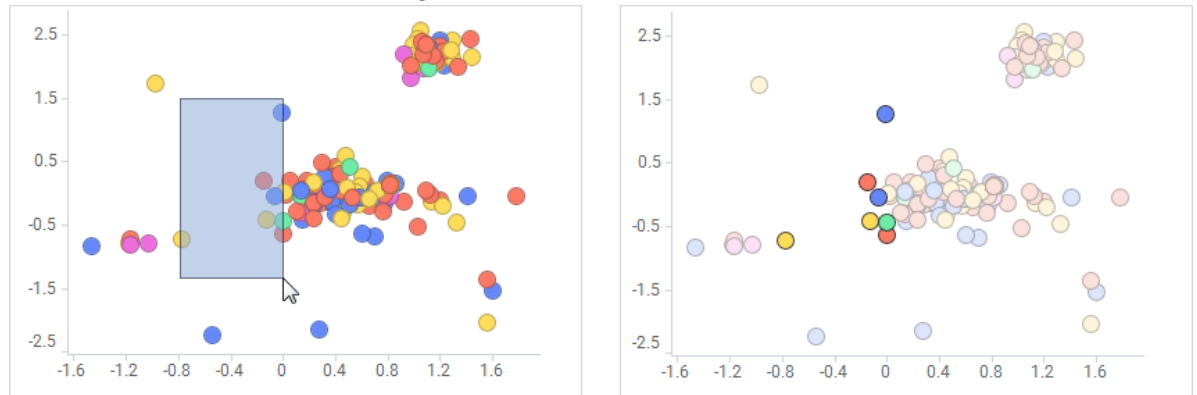
## Marking items

You can mark items in a visualization to view details for the items or to clarify connections between visualizations. Marking, for example, a pie sector or a bar is equivalent to marking all the data rows that are included in that pie sector or bar.

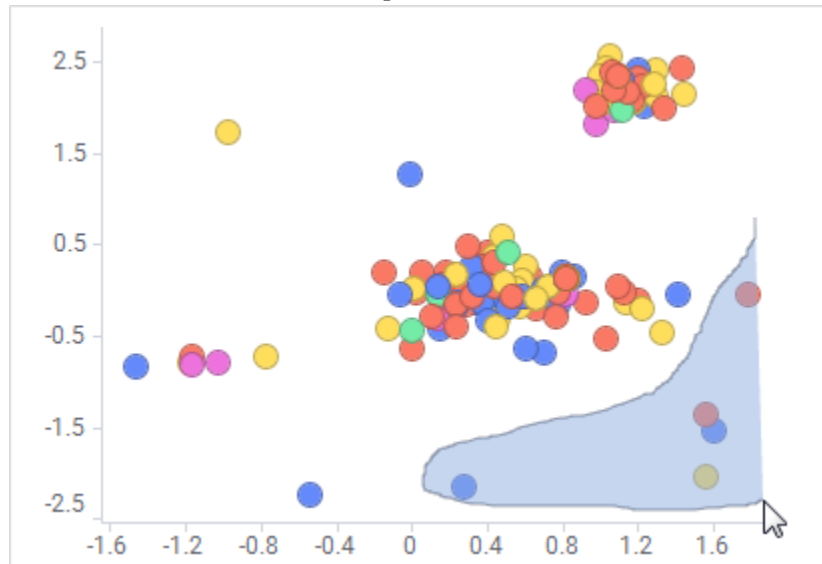
The following actions result in marked items:

- Click an item
  - marks the item and unmarks previously marked items
- Press Ctrl + click an item
  - adds or subtracts items from the marked set of rows

- Drag a rectangle
  - marks all items within the rectangle and unmarks all other items

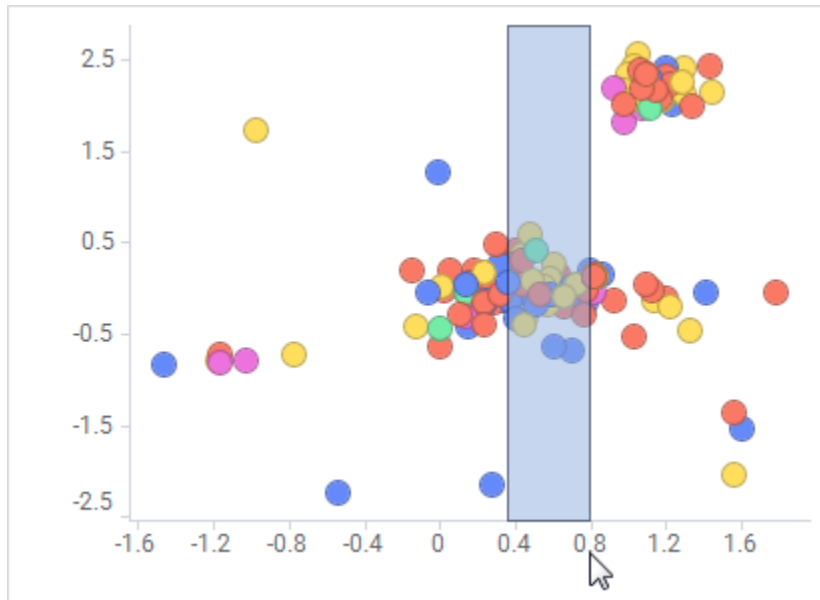


- Press Ctrl + drag rectangle
  - adds items within the rectangle that are not already marked to marked items
- Press Alt + draw a shape of any form (lasso marking)
  - marks all items within the shape



- Press Ctrl + Alt + draw a shape of any form
  - toggles items within the shape to the opposite of marked/unmarked

- Click the scale label area and drag along any of the axes
  - marks all items between the two axis values



Marking along the Y-axis in the bar chart is slightly different than normal rectangle marking; Only those bars with the top part of the bar included within the axis marking rectangle will be marked. This allows you to select bars with heights within a specific span.

- Select categories in the legend
  - marks all items of a certain category. For more information, see [Marking items using the legend](#).

## Marking items using the legend

You can mark items of a certain category using the legend in a visualization.

The legend displays settings made on various axes. An example of an axis is the **Color** axis, where each specified color represents a certain category. By clicking a category on an axis, you mark all items within that category.

For example, in the scatter plot below, the **Color** axis is used to mark all items in the "group d" category, that is, all yellow items. The other categories, "group a-c" and "group e", are grayed out in the legend to indicate that only the "group d" is marked.



The legend cannot be used to mark items on axes with numerical data. Only categorical data on the following axes can be marked: the color, shape and series axes.

### Procedure

1. In the legend, move the cursor over the axis category whose items you want to mark. A dotted line appears beneath the category if it can be used for marking.
2. Click a category to mark all its items.



Click the category text, not the symbol.

You can press Ctrl + click a category to mark more than one category. The categories can be selected on a single axis, or on different axes.

## Unmarking items

There are several ways to unmark a previously marked item.

- Unmark everything by clicking an empty area of a visualization.
- Right-click the visualization, click **Marked rows**, and select **Unmark**.
- Unmark marked items by pressing Ctrl while re-marking them.

## Reapplying markings when reloading data

When you are working with data that is linked to the source, you might need to consider what happens to your markings when the data is reloaded.

Spotfire tries to find a symbolic representation of the selection you make when marking data, whenever this is possible. That means, that if you mark an item in an aggregated visualization, the marking will be based on the selected categories, rather than on the underlying rows. However, symbolic representation is not possible to use with some visualizations or configurations. For example, if you mark rows in an unaggregated visualization or layer (for example, in a table visualization, or in a scatter plot with row number on one of the axes), or, if you use complex configurations (for example, categorical axes using OVER expressions in a hierarchy), then there will not be any symbolic categories available. Data of the data types Binary or Currency will also always use index-based marking for in-memory data. In these cases, you will lose the marking upon reload, unless you have specified [key columns](#) that can identify the separate data rows.



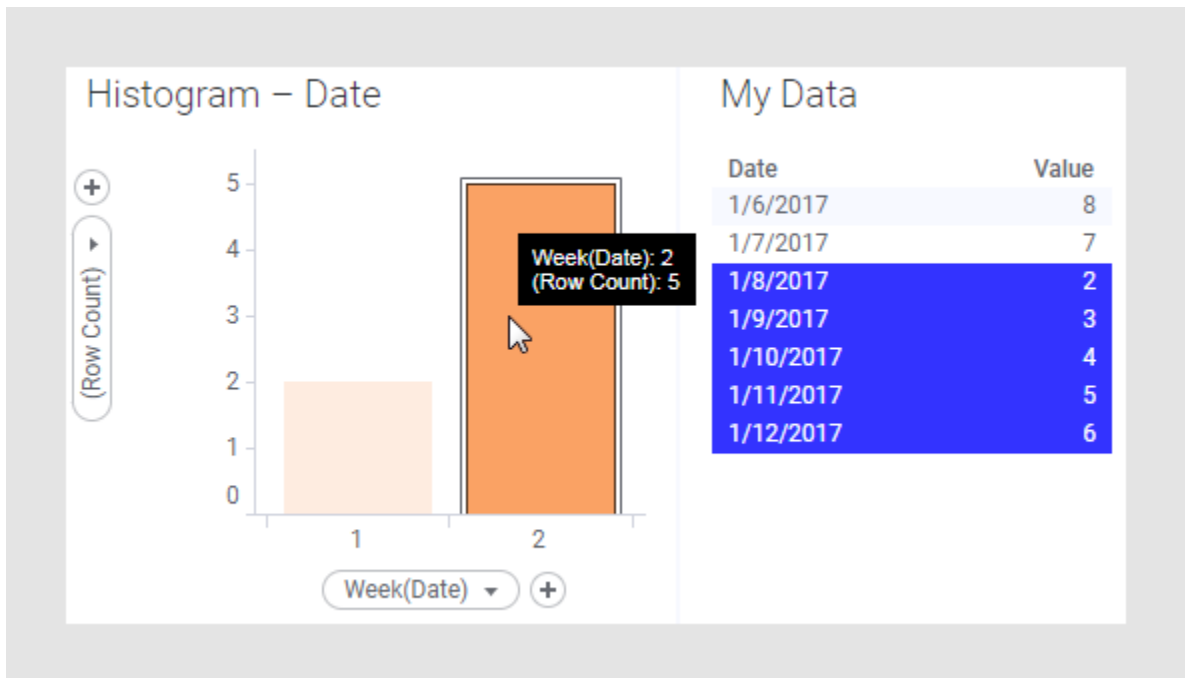
If you create analyses that should keep a marking after data has been [reloaded](#), for example, if you want to add a bookmark which marks some items in an analysis, then, you must make sure that either the interesting category, or the particular rows, are possible to identify even after the reload.

### What is important? The category or the distinct values?

For example, imagine that you want to include a bookmark where a specific category is marked. If the category is marked from a simple, aggregated visualization, like a bar chart based on categorical values from a column, and the data is reloaded to include more values within that same category, then all of the new values will be marked, because the symbolic marking is used. This way, you can easily configure a bookmark with a marking which always lets you see the data that currently is included in a particular category, and not the data that was included in that category when you first applied the marking.

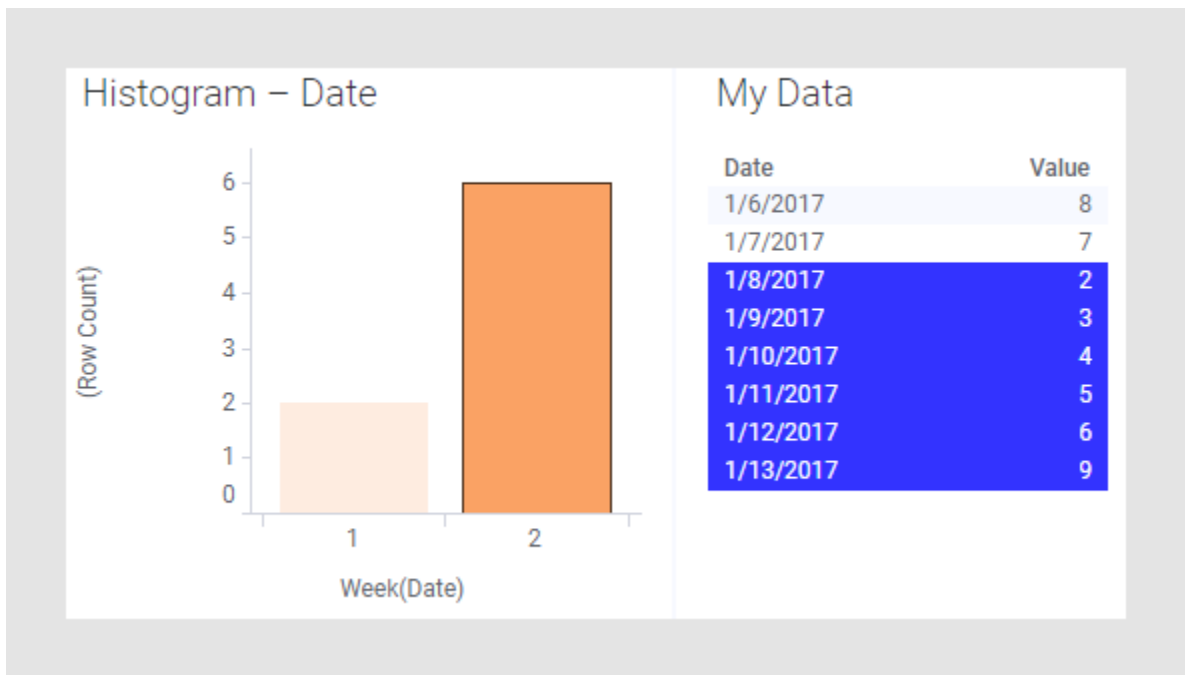
### Category

If it is the category that is of interest, you should mark the category in an aggregated visualization:



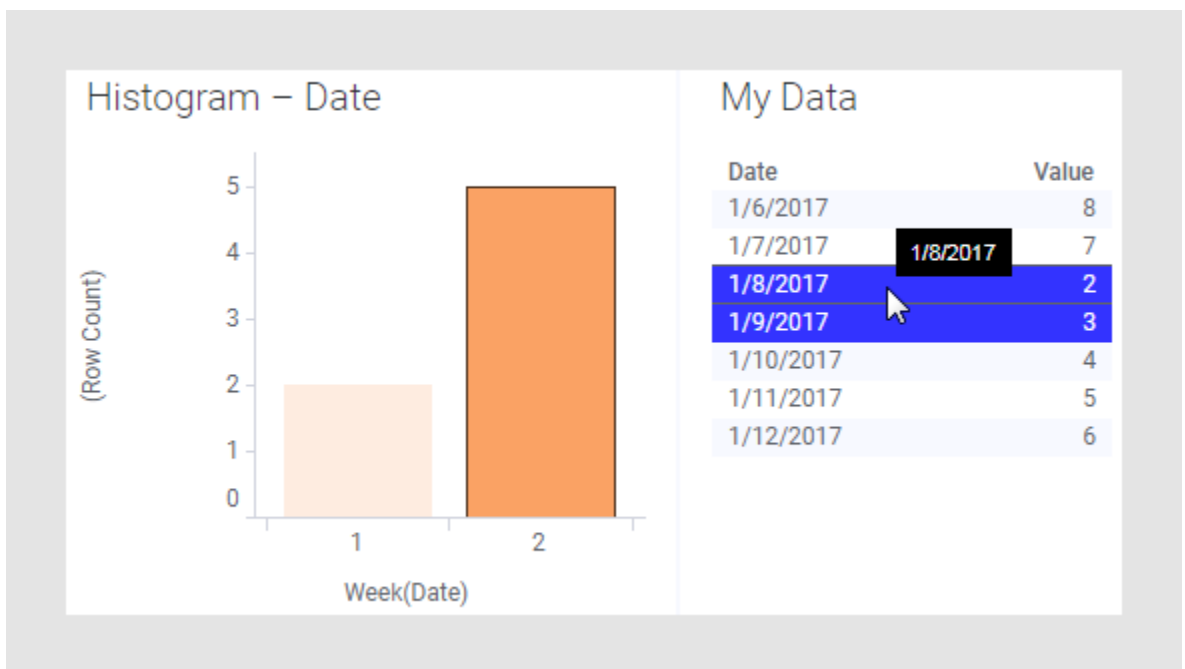
All rows belonging to that category will be marked, as shown in the table visualization using the same marking.

If the data table gets an additional row for week 2 and data is reloaded, then the new row is automatically included in the marked selection:

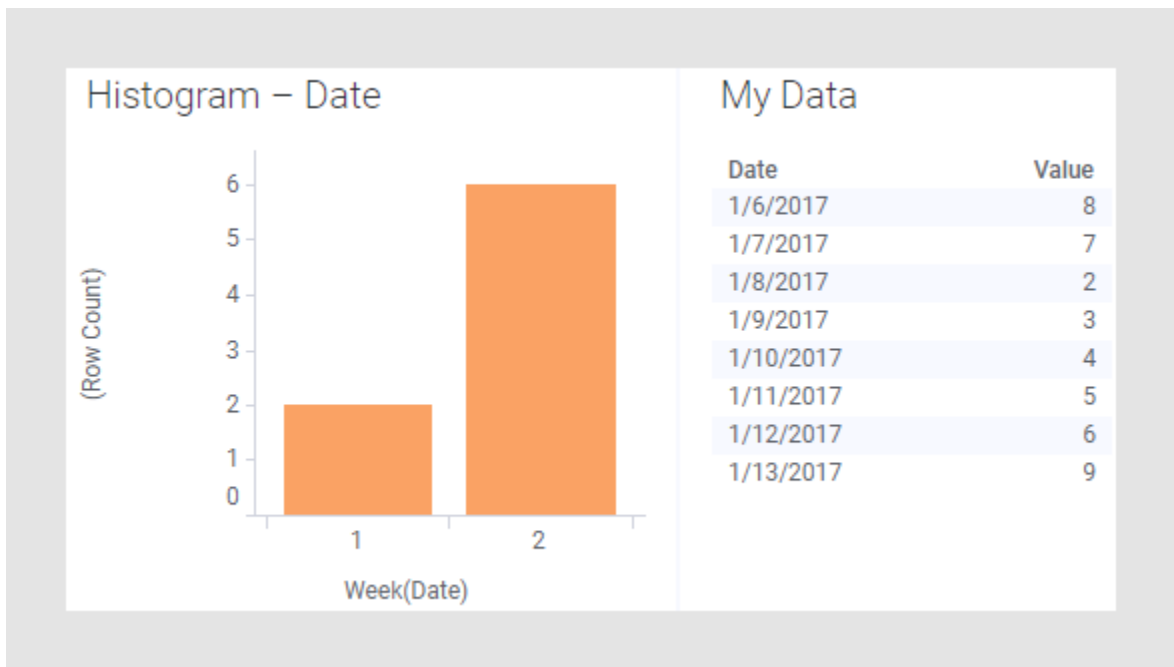


### Rows

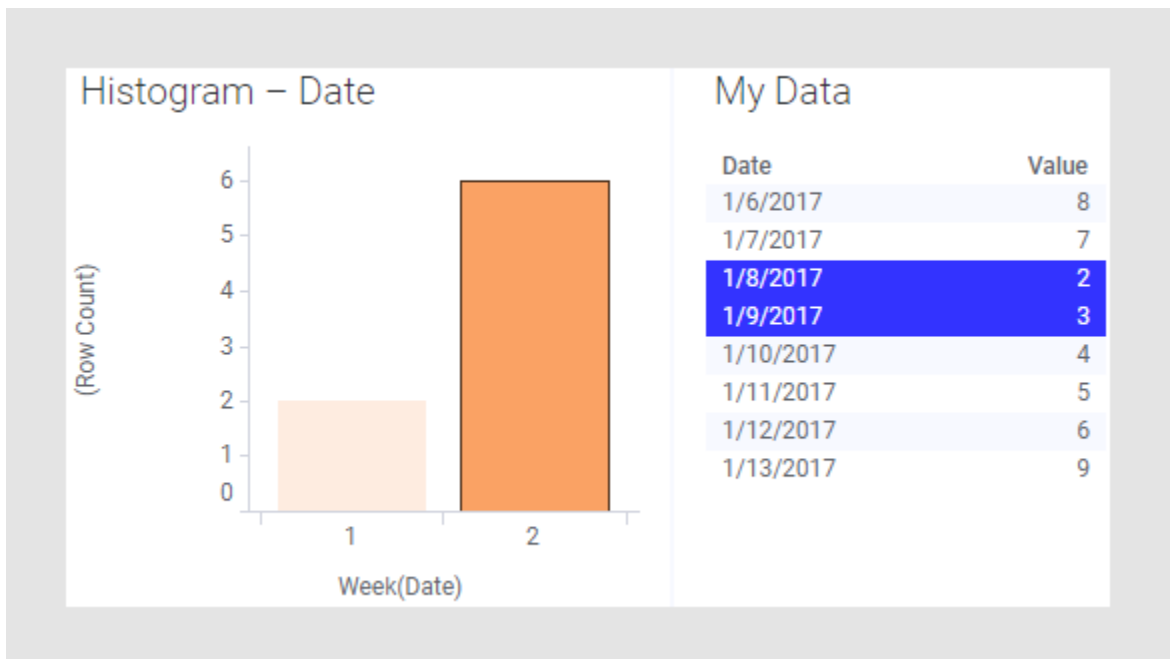
If it is the particular rows that are important, you make the selection in the table instead:



If no key columns were specified, a reload will make the marking disappear, because the rows could not be identified:

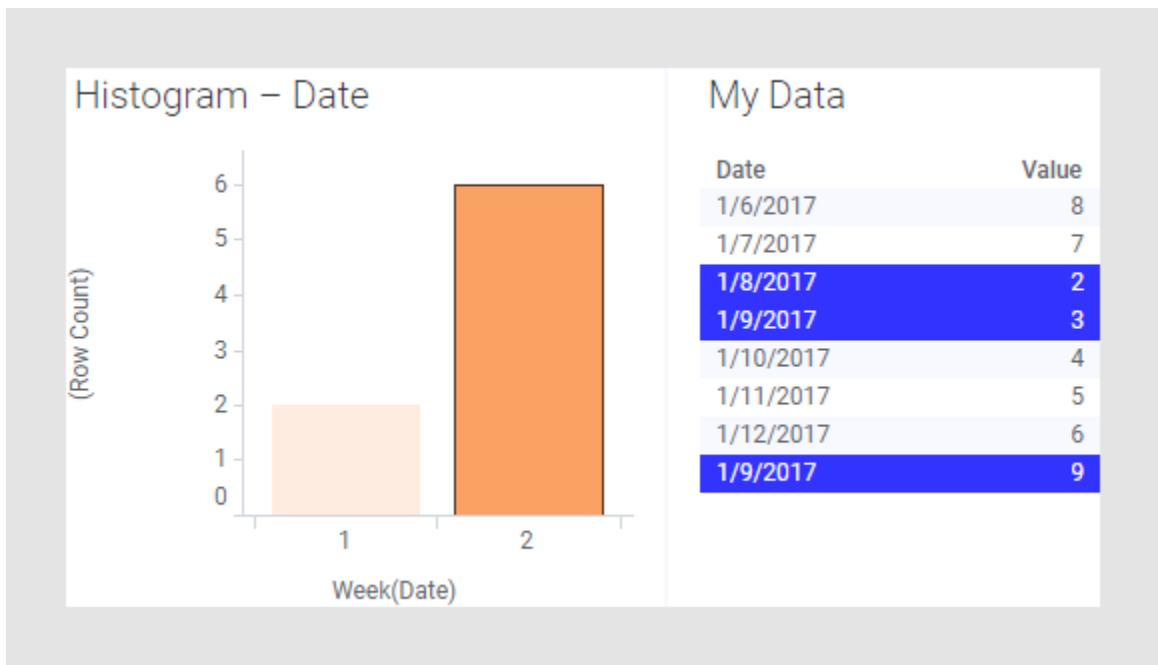


If key columns were specified and any changes to the data upon reload do not affect the key columns, then the marking will be kept, just as intended:



In this case, the Date column was used as the key column.

And finally, if the reload causes the key columns to break, so that they no longer uniquely identify all rows, you might end up in situations where more values are marked than expected:



In this example, another row with the same date was added to the linked data, and the previously unique key column was no longer unique. You might also end up in a situation where all markings are lost when the key columns are broken. You might be able to fix broken keys by adding more key columns in the [data canvas](#).

### External data

When you are working with external data (in-db) from a data connector, you will always use symbolic marking. However, if a primary key has been defined in the external data source, or, if a primary key has been specified in the Views in Connection dialog in Spotfire Analyst when setting up the data connection, it might still be possible to mark rows in a table visualization. See [Working with in-database data](#) on page 48 for more information about external data.

## Exploring data across visualizations

In an analysis, marking items in one visualization spreads to other visualizations based on the same data table. This way you can explore the same data in different contexts.

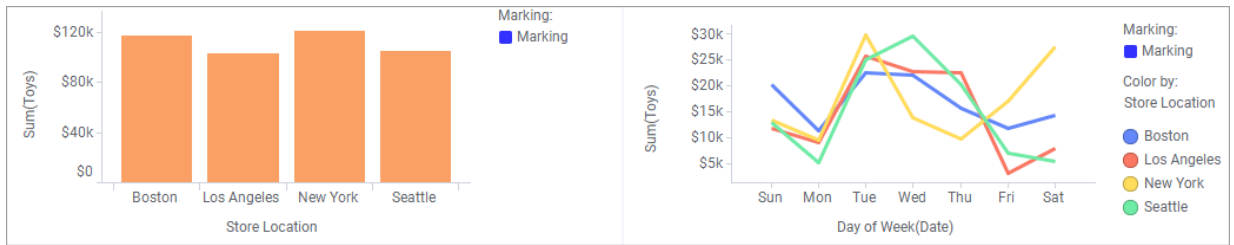
When you [mark items](#) in a visualization, you in fact mark the underlying data, that is, the data rows on which the marked items are based. The marked rows are associated with a specific *marking*, defined by a name and a color.

Other visualizations that include any of these marked rows are also affected. If any marked data row is part of an item in another visualization, this item also becomes marked. There is one condition though: to interplay, the visualizations must have the same *marking* applied.

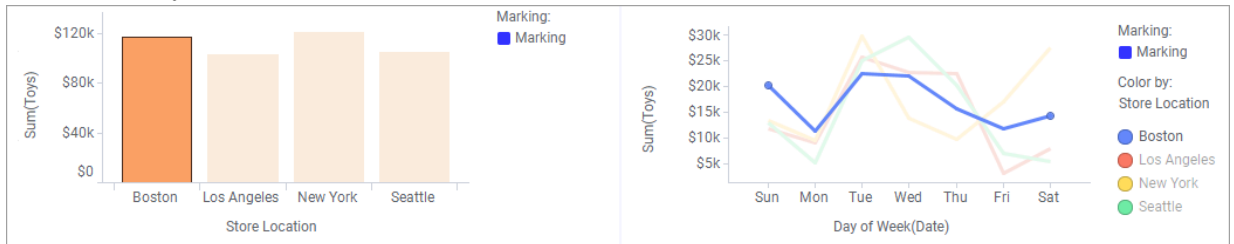
This interplay between visualizations can be used to follow up your marked data in another context.

### Example

The visualizations below show the sum of toys purchases from two perspectives. The bar chart presents the sum of purchases made at different store locations during a week, and the line chart presents these purchases per day.



If you mark any of the bars in the bar chart, corresponding line in the line chart becomes marked simultaneously.

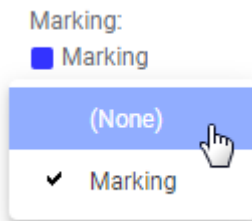


Moreover, you can easily switch between the store locations. Mark another bar, and corresponding line gets marked, alternatively mark another line, and corresponding bar gets marked.

The interplay is possible, because the visualizations are based on the same data table, and the same *marking* is applied. The table with the marked rows is shown below.

Store Location	Date	Toys
Los Angeles	7/21/2016	11266
Seattle	7/23/2016	185
Boston	7/23/2016	354
Boston	7/23/2016	0
New York	7/23/2016	18918
New York	7/23/2016	304
Los Angeles	7/23/2016	0
New York	7/23/2016	0
Los Angeles	7/23/2016	0
Boston	7/24/2016	1934
Boston	7/25/2016	91
New York	7/25/2016	963
New York	7/26/2016	499
New York	7/26/2016	0
Seattle	7/26/2016	1092
Seattle	7/26/2016	122
Seattle	7/26/2016	963
Los Angeles	7/26/2016	0

In case you do not want the marking of items in a visualization to affect another visualization, select the (None) option in the **Marking** drop-down list in the legend.



## Adding a new marking

If you want some marking to be specific to one or more visualizations, rather than affecting the entire analysis, you can add one or more markings. In the Markings panel, you can also edit or delete markings, or get an overview of where different markings are used.

### Prerequisites

You must be an author with the Simple or Advanced Visualization Properties license feature. The analysis must be in **Editing** mode.

### Procedure

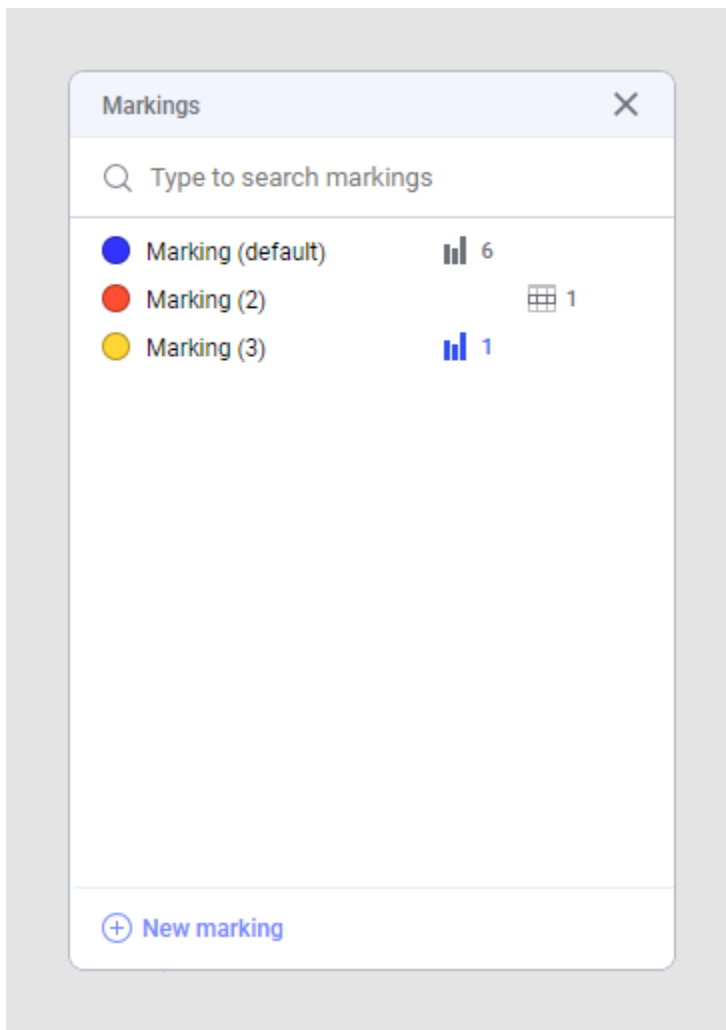
1. On the menu bar, select **View > Markings**.
2. Click **New marking**.

### Result

The new marking is added to the list, and it can be used as the marking, or as a data limiting marking, for any visualization. These settings are specified in the visualization properties. If the marking is shown in the legend, you can also change the marking to use from there.



You can click on the color icon and change the color of a marking at any time, or double-click on the name of the marking and change its name.

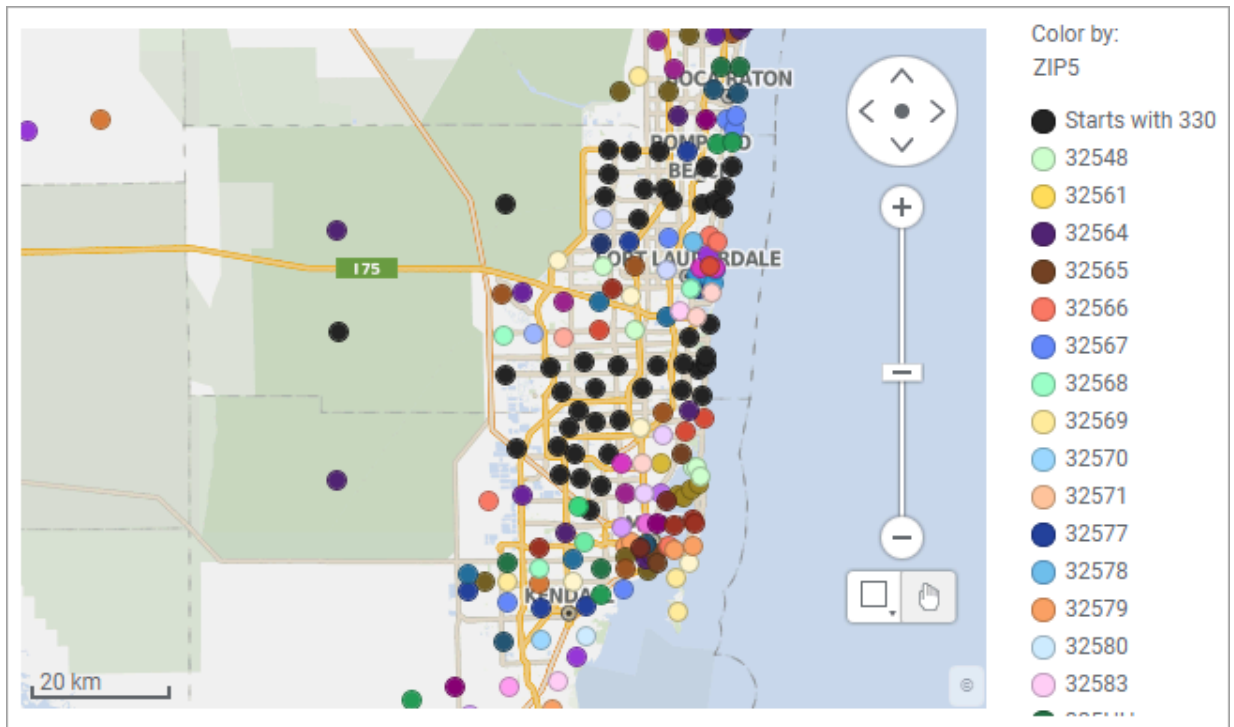
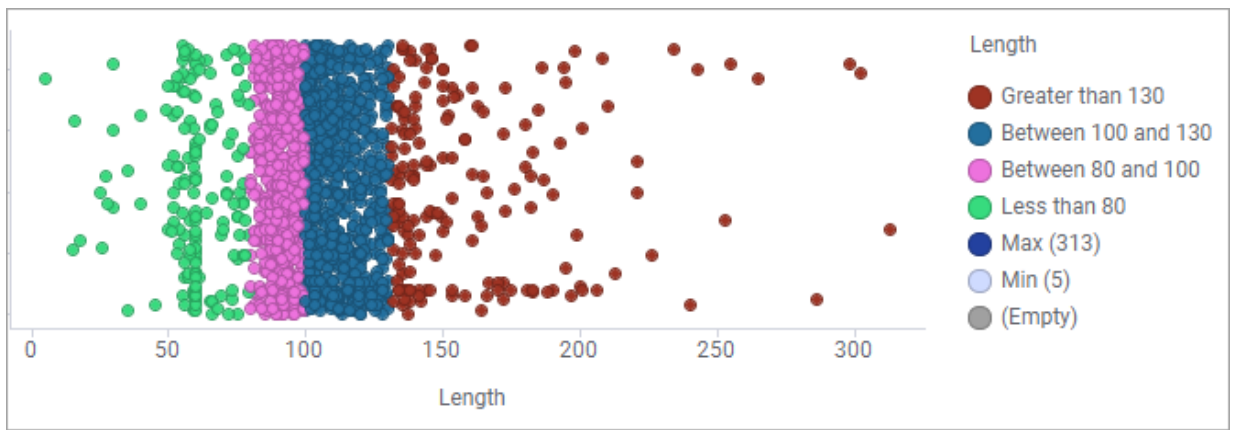


## Working with colors

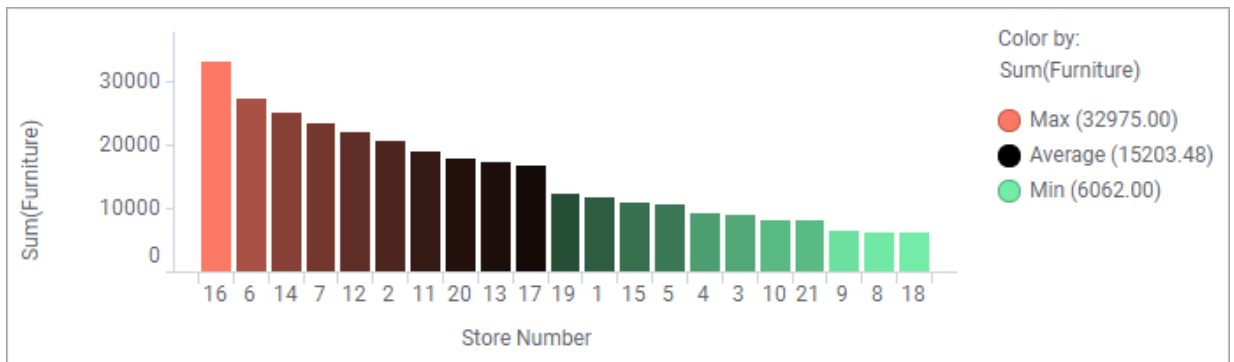
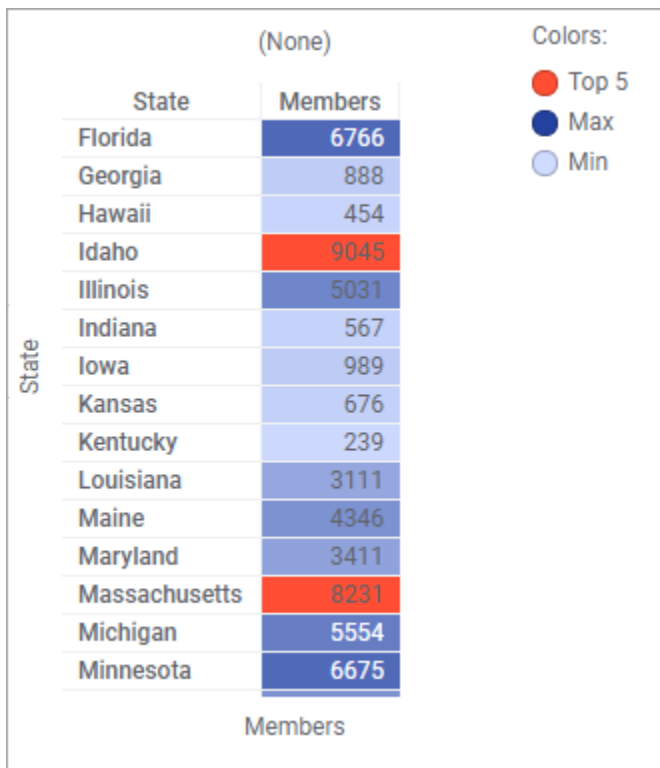
Colors can be used to add further information to a visualization. You can color items to, for example, draw attention to certain items, distinguish particular categories of data, present variation in values, or identify outliers.

On the **Color by** axis in the visualization, you select the column whose values you want to color the items by. The column can be any of the columns that are already selected on any other axis in the visualization, or totally different ones. Depending on whether the selected column contains categorical values or numerical values, you have different coloring options.

Examples of how colors can add more information to visualizations are shown below. You find the details of the coloring in the legends.





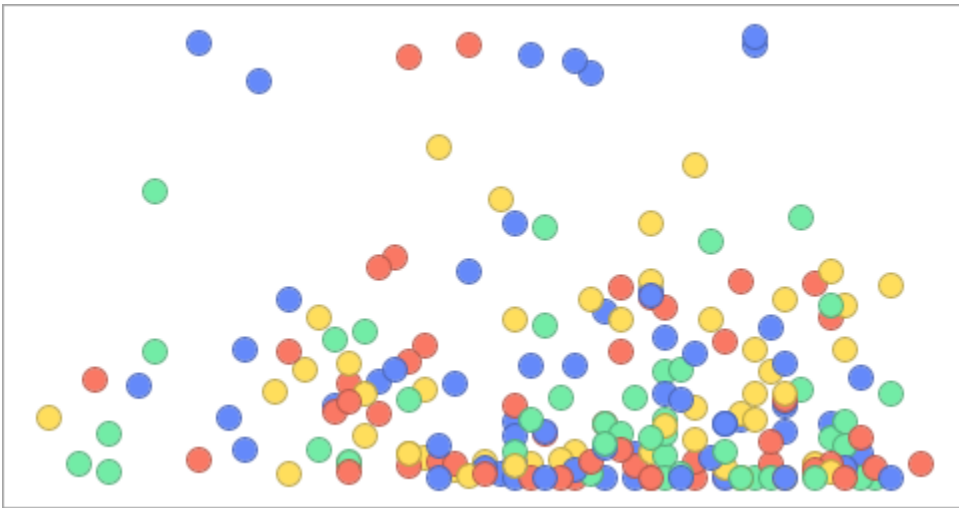


Which color is assigned to a certain value can easily be changed. Simply click the colored circle in the legend, and pick another color in the palette that is displayed.

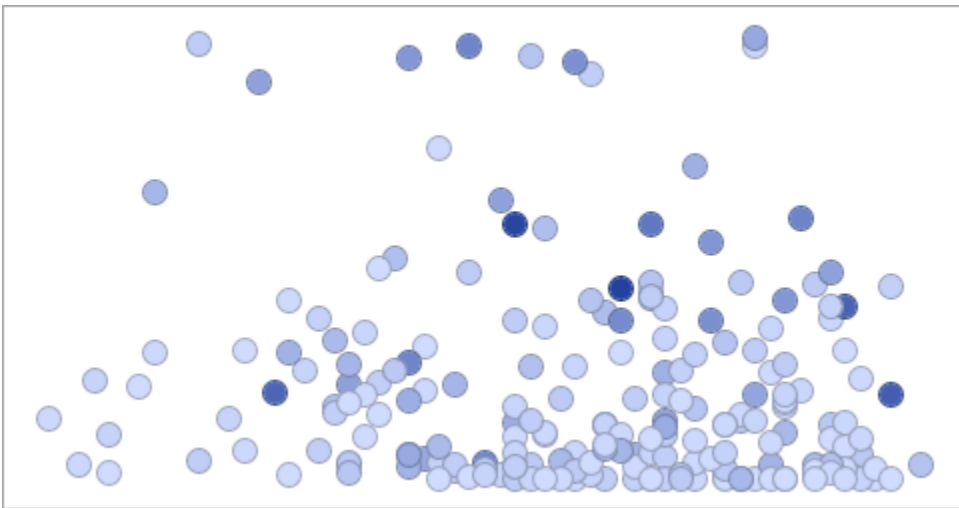
### Distinct colors and gradient coloring

Different coloring possibilities are provided depending on whether the selected column on the **Color by** axis contains categorical values or numerical values. Categorical values are values that can be divided into groups such as gender or country, and numerical values are quantitative data such as sales figures or distances.

- When you use categorical data on the color axis, all items within a category are given a common, distinct color.



- When you use numerical data on the color axis, a gradient color transition reflecting the varying quantities can be applied to the items.



### Color rules

It is possible to make deviations from the rest of the coloring. For example, you may want top value items stand out using a certain color, or values with identical start characters, or you may want to color items in a certain interval differently. To do so, you [specify rules containing conditions for the coloring](#). Items that match your condition are colored according to your rule instead of following the rest of the coloring pattern. It is possible to add several color rules to a visualization.

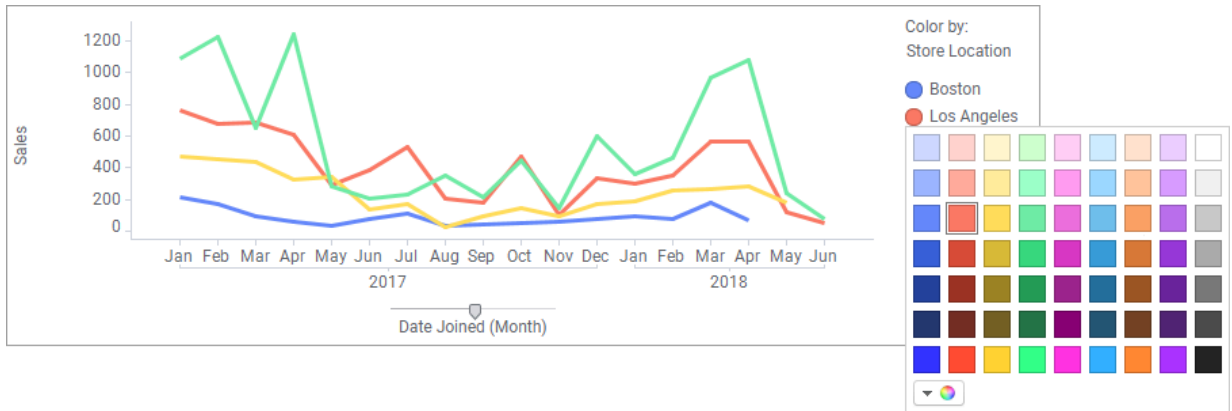
The top image shows some examples of visualizations where color rules have been specified.

### Color schemes

Finally, you can color the visualization items [using a predefined color scheme](#). Which predefined color schemes are available differs depending on type of data values in the column. Some of the color schemes have built-in color rules. You can use a predefined color scheme as it is, or use it as starting point for your adjustments.

## Changing color of visualization items

The Color by axis specifies how to color items in a visualization. The colors can easily be changed. The **Color by** axis setting is available in the [legend](#) of the visualization.



### Procedure

1. In the [legend](#) beneath the **Color by** column selector, click the circle for the color you want to change.



In combination charts, the coloring is specified on the **Series by** axis.

A color palette is displayed.

2. Select another color on the palette.

You can click  for more color options.



The color palette is available also in the **Color** section of the visualization **Properties**.

## Adding color rules

By adding color rules, you can let the colors of specified items deviate from the ordinary coloring of the visualization items.

A color rule contains a condition and a color. All visualization items that fulfill the condition get the color you have specified. For example, you can add rules that let the top values, or values in a specified interval, stand out in a certain color. See some examples in [Working with colors](#).

To specify conditions, different rule types are predefined. Which rule types are available depends on whether the column that is selected on the **Color by** axis contains categorical values or numerical values.

More than one color rule can be added to a visualization.

### Procedure

1. Right-click the visualization, and select **Properties** in the opened menu.  
The Properties pop-over is displayed.
2. Click **Color**.  
The **Color** section is displayed.
3. Make sure the column you want to color by is selected on the **Color by** axis.
4. Click **Add color rule**.  
The Add color rule dialog is displayed. The dialog provides different rule types depending on whether the selected column on the **Color by** axis contains categorical values or numerical values.

5. Select a **Rule type** in the drop-down menu.

Different rule types require different inputs such as fixed values that you type, aggregations selected from the drop-down menu, or [custom expressions](#). For example, in a **Top** rule, the input is how many top values should stand out.

6. In the field below **Rule type**, specify the needed input using the drop-down menu or entering a value.
7. Click the **Color** square to, from the displayed palette, select which color to use for the items fulfilling the rule.
8. If you want, type a **Display name** for the added rule in the field.  
The display name is displayed in the legend. If you leave the field empty, the default rule type name is shown.
9. Click **OK**.

### Result

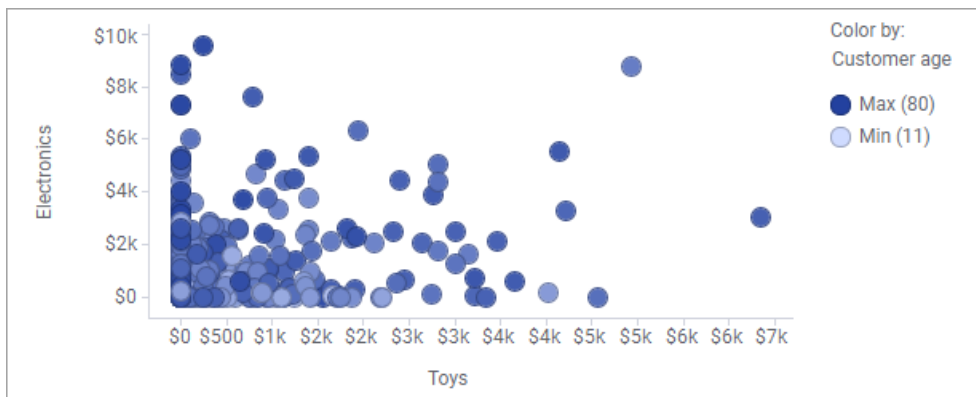
The color rule is applied to the visualization.



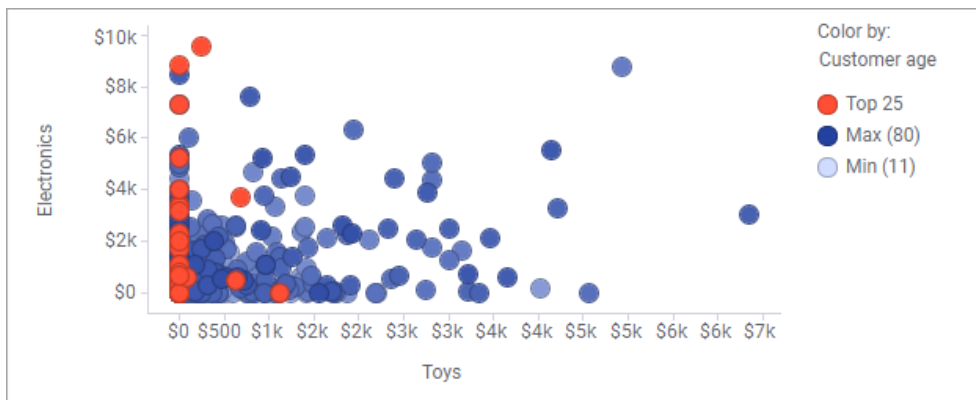
More than one color rule can be added to a visualization. The most recent color rule is added on top of the others. However, the [order in which the color rules are applied can be changed](#).

### Example of color rule

In the scatter plot below, the markers represent how much money each customer spent at the electronics and toys departments in a store. The gradient coloring of the marker indicate the age of the customers.



Assume you want to examine the buying behavior among the oldest customers. By adding the **Top** color rule with 25 as input value, the markers representing the 25 oldest customers are distinguished.



In the scatter plot above, also the **Drawing order** is specified to Customer age to draw the lowest values first in the plot, and the highest values last. This means all the markers of interest in this particular example are drawn on top of the others, and therefore not hidden behind others.




Color rules can be part of **color schemes**. For example, the **Top** rule used above is in fact the **Top n** color scheme. The default n value in the color scheme though is 10, but it can be changed easily.

### Editing color rules

Existing color rules can be edited.

#### Procedure

1. Right-click the visualization, and select **Properties** in the opened menu. The Properties pop-over is displayed.

2. Click **Color**.  
The **Color** section is displayed.
3. Beneath **Color items**, click  for the color rule to be edited.  
The Edit color rule dialog is displayed.
4. [Adjust the color rule](#).

## Changing order of color rules

A visualization can contain several color rules. You can change in which order they are applied.

The order, which can be viewed in the legend, is important, because an upper [color rule](#) takes priority over a lower rule.

### Procedure

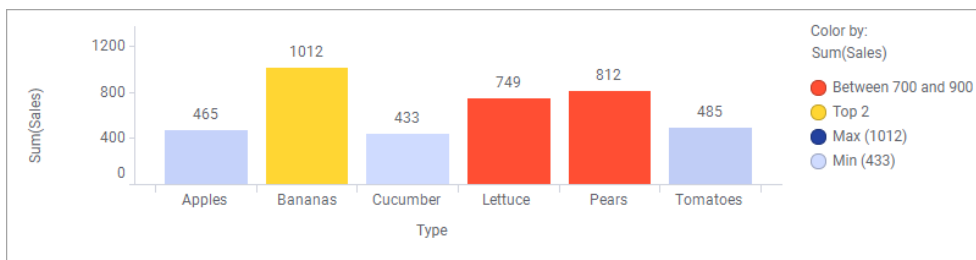
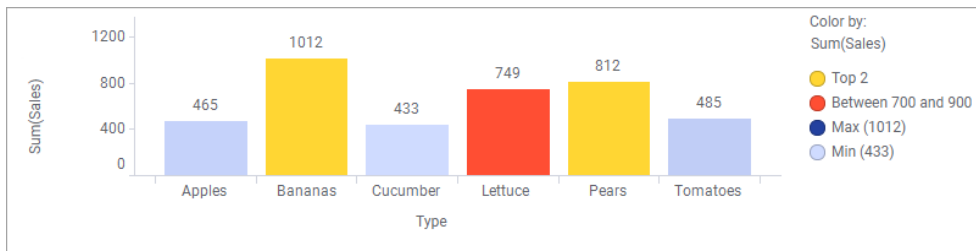
1. Right-click the visualization, and select **Properties** in the opened menu.  
The Properties pop-over is displayed.
2. Click **Color**.  
The **Color** section is displayed.
3. Beneath **Color items**, click and drag the color rule to the new position.

### Result

The new order of the color rules is applied.

### Example of rule priority

The bar charts below show the sales of different fruits and vegetables. In both the charts, two color rules are added. One color rule is defined to display the two highest bars in yellow, and the other color rule to display bars between 700 and 900 in red. Note the different orders of the color rules, and the different coloring.



One bar, 'Pears', fulfils the conditions in both the rules. However, in the first bar chart, the 'Top 2' rule overrides the 'Between 700 and 900' rule, and in the second bar chart, the opposite priority is applied.




If a color rule does not have any affect on the visualization, it is not listed in the legend.

## Removing color rules

You can remove a color rule.

### Procedure

1. Right-click the visualization, and select **Properties** in the opened menu.  
The Properties pop-over is displayed.
2. Click **Color**.  
The **Color** section is displayed.
3. Beneath **Color items**, click  for the color rule to be deleted.  
The color rule is deleted, and no item colors deviate from the ordinary coloring pattern.

## Adding color schemes

You can color visualization items using a predefined color scheme. A color scheme can be used as it is, or used as starting point for any adjustments of the coloring.

Which predefined color schemes are available differs depending on type of data values in the column selected on the **Color by** axis. Some of the color schemes have built-in color rules.

### Procedure

1. Right-click the visualization, and select **Properties** in the opened menu.  
The Properties pop-over is displayed.
2. Click **Color**.  
The **Color** section is displayed.
3. Make sure the column you want to color by is selected on the **Color by** axis.
4. In the **Color scheme** drop-down menu, select the color scheme you want to use.



Any color rules in the applied color scheme are displayed in the legend.

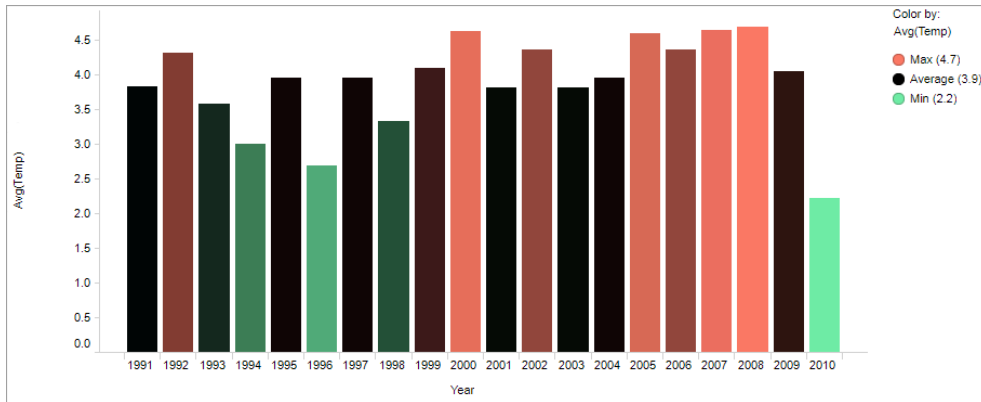


**Accessible categorical** is a color scheme for categorical columns with a palette that is compliant with the WCAG 2.1 and conforms with Level AA or higher for the contrast between text and background. There are two variants, one adapted for the light theme and one for the dark theme.

### Result

The color scheme is applied.

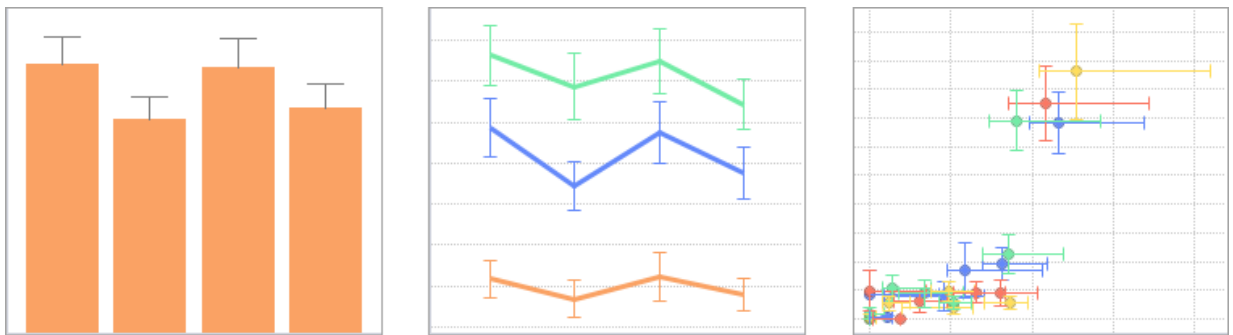
## Example of color scheme



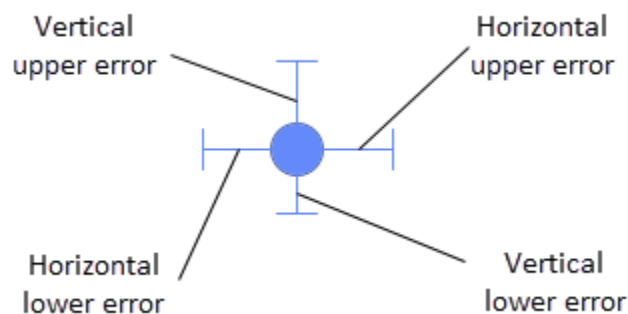
The bar chart shows yearly average temperatures for a 20-year period. Not only the bar heights reflect the temperature values but also the bar colors. The **Min-avg-max** color scheme is applied. It represents values from Min to average with a gradient transition from green to black. Items representing values from average to Max are colored in a gradient transition from black to red.

## Error bars

Data might not always be precise, for example, in measurement data error margins might exist. When there is a need to visualize the uncertainty in the data, you can use error bars to indicate statistical probabilities of errors, or actual errors. The error bars represent upper and lower limits of the data relative to a marker's value, and they can be added to markers in bar charts, line charts, and scatter plots.



In all these three types of visualizations, upper and lower vertical error bars can be drawn. In scatter plots, the error bars can also be drawn horizontally.





If more than one column is selected on the axis selector, you can specify different error bars for each of the columns.



When you work with error bars in bar charts, make sure that the bar chart is displayed using side-by-side bars.

When specifying what the length of the error bars should indicate, you have different options. You can

- let the length represent the actual values in a data column that contains absolute error figures.
- for aggregated visualization items, use any of the pre-defined aggregation methods, for example, standard error (StdErr) or standard deviation (StdDev).
- define your own method by writing an expression.

This means that you can use error bars also for other purposes than indication of data uncertainty.



Error bars are drawn relative to a marker. For example, in a bar chart, an upper error bar will be drawn with the top of a bar as starting-point.

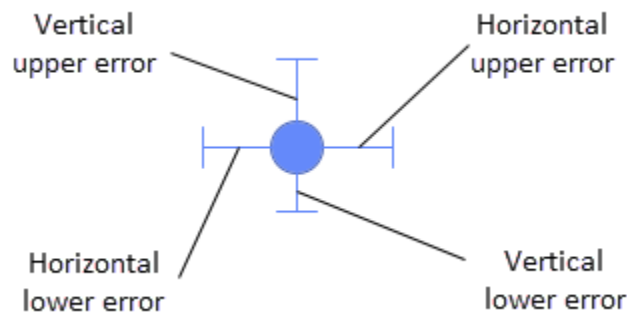
For more information on how to use the error bars, see [Adding error bars](#).

## Adding error bars

Typically, error bars are used to visualize uncertainty in the data. You can add them to visualization items in bar charts, line charts, and scatter plots. The degree of how uncertain an item value is, is indicated by the length of the error bar.

In fact, you can use [error bars](#) to indicate also other types of ranges than uncertainty in values, because you have various options to define what the error bar lengths should represent. The examples following the procedure show different use cases.

In all the three types of visualizations, where error bars can be added, upper as well as lower vertical error bars can be drawn. In scatter plots, the error bars can also be drawn horizontally.



### Procedure

1. Right-click the visualization, and select **Properties** in the opened menu.
2. In the Properties pop-over, click **Error bars**.
3. In the Error bars section, under **Visible error bars**, select **Vertical** to draw vertical error bars, and **Horizontal** to draw horizontal error bars.  
The **Horizontal** check box is available only in scatter plots.
4. Use the **Upper error** and **Lower error** selectors to specify what type of range to display. See the examples below for different options.

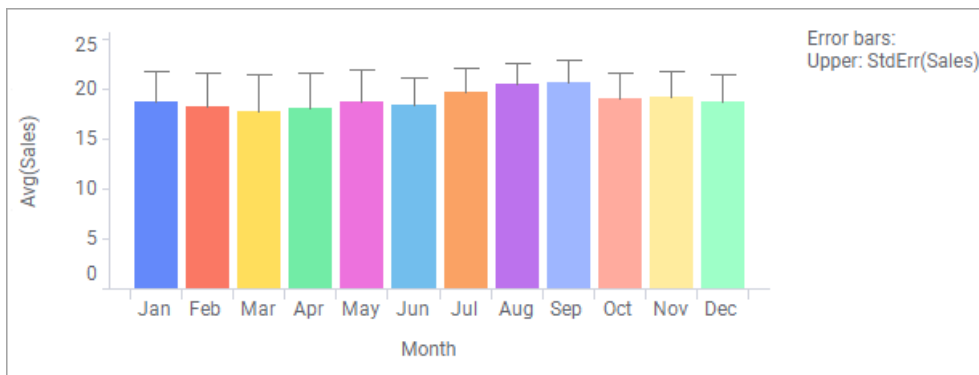


The error bars are by default drawn relative to the marker position in the visualization.

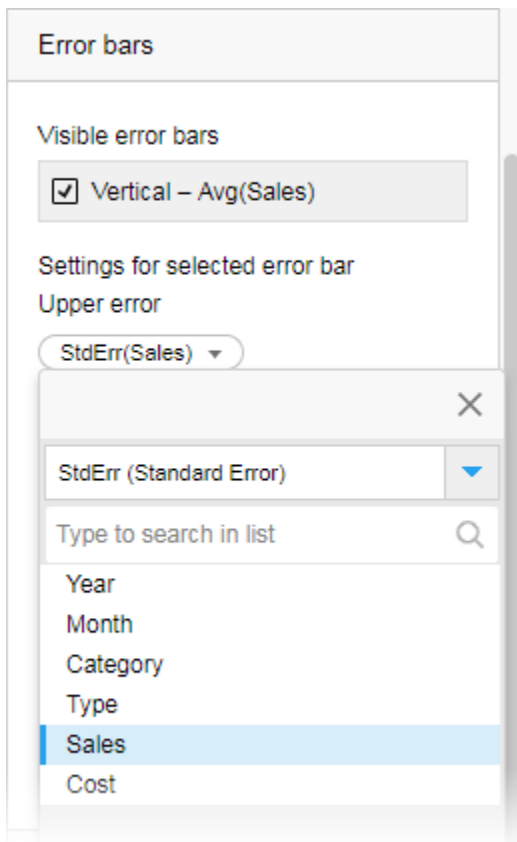
5. Specify the **Color** of the error bars:
  - Select **Same as marker** to let an error bar have the same color as the related marker.
  - Select **Custom** to access a palette to specify a different color.
6. Select whether to **Show end caps** of the error bars.
7. Select **Include error bars in axis range** to always show the entire error bars. This means that the range of the axis scale is extended automatically in the visualization to make the error bars totally visible.

## Error bars based on an aggregation

The bar chart shows the average sales per month during one year. To show the uncertainty in the values, the pre-defined aggregation method standard error (StdErr) has been used to calculate the length of the error bars.



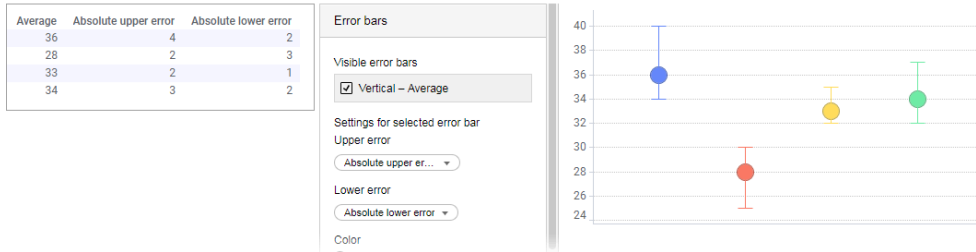
The setting is shown below. The StdErr aggregation is specified on the **Upper error** selector.



A long error bar in this example means that the concentration of values, which the sales average was calculated on, is low, and thus the average value is uncertain. Conversely, a short error bar means that the concentration of values is high, and thus, that the average value is more certain.

## Error bars based on absolute column values

The data table below lists already calculated averages and estimated upper and lower errors in absolute values. By, for the 'Average' axis, setting the **Upper error** to the 'Absolute upper error' column and the **Lower error** to the 'Absolute lower error' column in a scatter plot, the error bars for each average value represent the actual values in the two error columns.



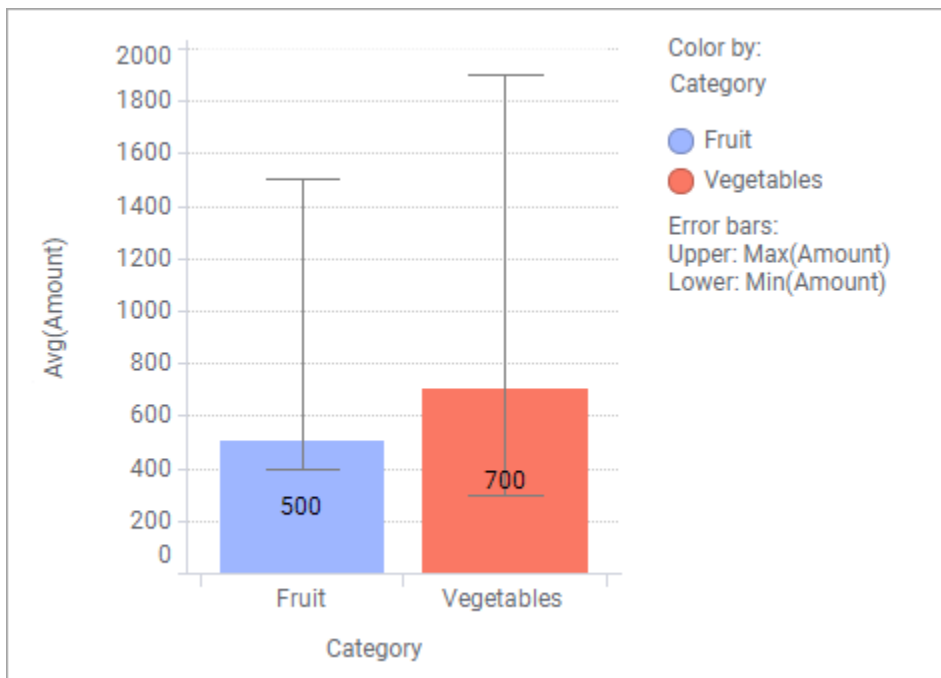
### Error bars based on a custom expression

Instead of using a pre-defined aggregation method, you can specify the length of the error bars by writing your own [expression](#). This is an example where custom expressions are used.

This simple data table lists eight sales transactions.

Customer ID	Category	Amount
ID1	Fruit	100
ID4	Fruit	300
ID3	Fruit	500
ID4	Fruit	1000
ID5	Fruit	600
ID1	Vegetables	400
ID2	Vegetables	500
ID3	Vegetables	1200

Assume you are interested in visualizing the average sales amount per category in a bar chart, and also in showing between which amounts the transactions vary by means of error bars. Then a custom expression is needed for the error bar calculations. If you directly use the pre-defined Max() and Min() aggregation methods for the amount column as shown below, the variation is drawn incorrectly. It should, for example, be 100-1000 for fruits instead of 400-1500. This is because the error bars are always drawn relative to the marker, meaning the upper end cap equals the sum of the average amount and the max amount (for example, 700+1200 for vegetables).

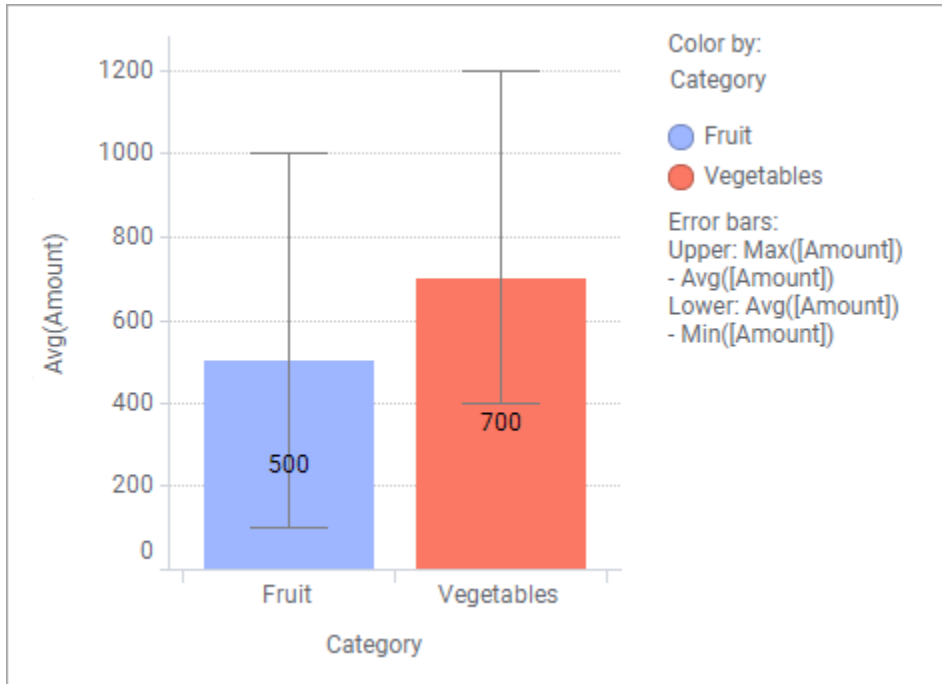


To show what you intended, right-click the **Upper error** selector, select **Custom expression**, and enter the expression that subtracts the average value from the max value:

```
Max([Amount]) - Avg([Amount])
```

In the same way, for the **Lower error**, enter the expression:

$\text{Avg}([Amount]) - \text{Min}([Amount])$



Now the error bars in the bar chart show between which amounts the transactions vary.

## Information and warnings

If something is off, you might see information or warning icons in different places in the analysis. You can often hover with the mouse pointer or click the icon to see more information.

### Information icons in visualizations

If there are items in a visualization that cannot be shown or might be misinterpreted because of the current settings, an icon is shown on the title bar of that visualization. If something cannot be shown in the visualization, a notification icon ⓘ, is displayed. If there is a risk that something might be misinterpreted due to lack of data to calculate a curve or similar, a warning icon ⚠, is shown. On mouseover, a tooltip is shown, and if you click the icon a longer description is displayed.

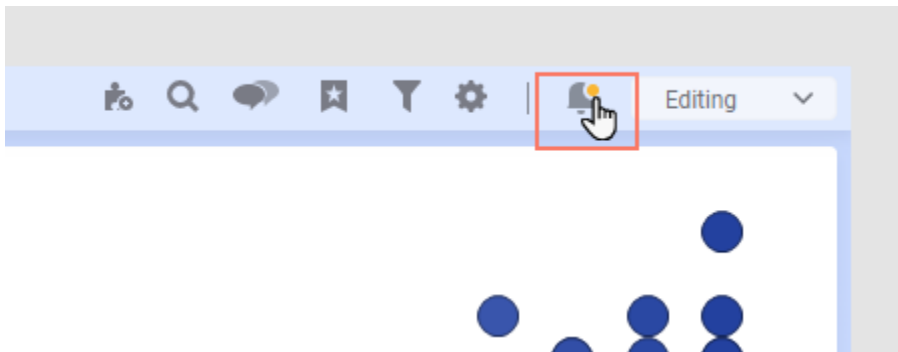
### Getting system information

You can see information about the server and deployment area you are connected to by hovering with the mouse pointer on the globe icon to the right on the status bar 🌐. There, you can also see the user you are logged in as, and which client or server version you use.

Click the globe icon to see progress information for calculations and processes that run in the background.

### Viewing details about notifications and warnings

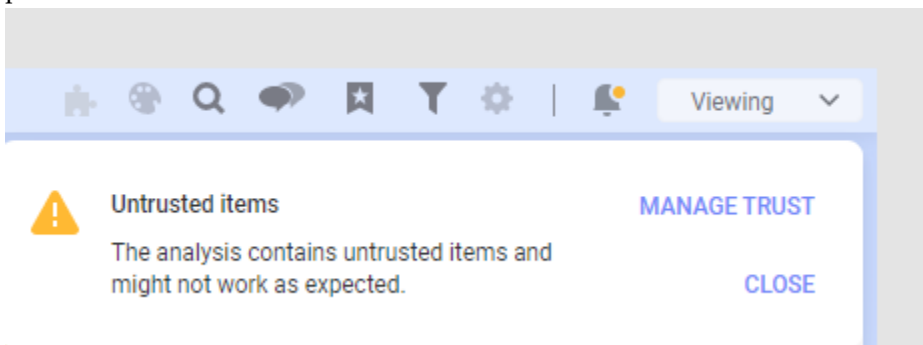
If there is some trouble with the loaded data, with a script or data function, or with a calculation in the analysis, a warning indication is displayed on the notifications icon on the menu bar. If pop-up notifications are enabled then new notifications will temporarily show up automatically.




Click the notifications icon to display all notifications about current issues in a panel.

Click on a specific notification to view more details about that particular issue in a separate dialog. From this dialog, you can copy alerts to the clipboard and use the information when communicating with others during troubleshooting of an analysis.


If the notification requires an action and you have the correct licenses to act on it, you might see a button taking you to the place where the action is needed. For example, if you open an analysis file where there are untrusted scripts or data functions and you are a script author, you can click the **Manage trust** button to go to the [Manage trust](#) dialog and review and trust the script to solve the problem.



### Switching off pop-up notifications

1. Click the **Notifications** icon on the toolbar .
2. In the Notification panel, clear the **Show pop-up notifications** check box.

### Clearing the notifications panel

1. Click the **Notifications** icon on the toolbar .
2. In the Notifications panel click **Dismiss all**.

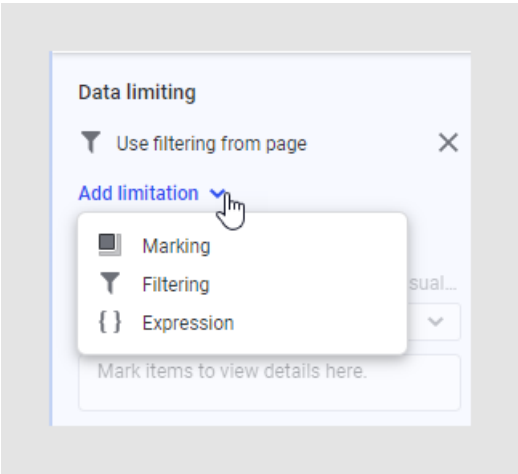
## Adding data limitations for a visualization

The default behavior when adding a new visualization is that it is automatically updated with the filtering on the page. However, it is possible to change the data limiting so the visualization is not changed by filtering at all, or so that it uses another filtering scheme. You can also make the visualization respond to one or more markings in other visualizations (often called details visualizations). Other options to limit the data is by using expressions, or, for streaming data, by

specifying time ranges. You can change the data limitation settings under **Data** in the **Visualization properties**.

**Procedure**

1. Click to select the visualization to limit.
2. Open the visualization properties and locate the **Data** section.
3. Under **Data limiting**, click **Add limitation** and choose which type of limitation to use.



For some visualizations, you can add a data limitation that only applies to a specific part of the visualization. For example, layers in a map chart, KPIs in a KPI chart, or columns in a graphical table. In the visualization properties, select the layer, KPI, or column of interest, click **+ Add**, and select **Item data limiting**.

Option	Description
Marking	<p>Choose one or more of the available markings to limit what is shown in the visualization. This means that the visualization only shows data that has been marked in other visualizations (main visualizations), or has been marked within the visualization itself.</p> <p>If more than one limiting marking is selected, you can choose how the data in the markings should be combined in the popover next to <b>Settings for limiting markings</b>. Use <b>All markings (AND)</b> to show the intersection of markings, and <b>Any marking (OR)</b> to show the union of the selected markings.</p> <ul style="list-style-type: none"> <li>• <b>All markings (AND)</b> – Use this if you have two main visualizations with different markings and want to see which markers are present in both markings. This visualization will then show only the data that has been marked using both of the selected markings.</li> <li>• <b>Any marking (OR)</b> – Use this if you have two main visualizations with different markings and want to see which markers are present in either of the two markings. This visualization will then show data that has been marked using either of the selected markings.</li> </ul>

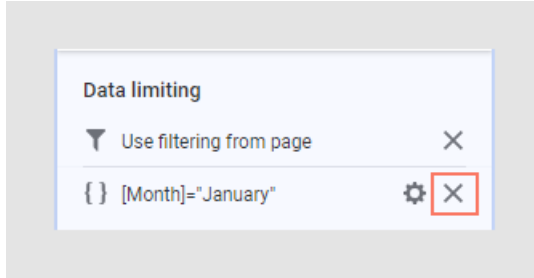


Option	Description
	<div data-bbox="908 289 951 338" data-label="Image"></div> <p>You can specify what should be shown in the visualization if no items have been marked in the main visualizations using the <b>If no items are marked in the limiting visualizations, show</b> drop-down list. For example, you can add a message with instructions.</p> <p>If you want to be able to mark data and show it within the visualization itself, the <b>All data</b> option must be selected in this list.</p> <p>See <a href="#">Drilling down into details</a> on page 597 to learn more about how you can use details visualizations to drill down into your data.</p>
<b>Filtering</b>	<p>Defines how different filtering schemes in the analysis should affect this visualization.</p> <p>Select <b>Use the current filtering from the page</b> if you want the visualization to always respond to the filtering scheme that is used on the page where the visualization is located. If you move the visualization to a new page, then the visualization will automatically start reacting on the filtering scheme that is used on the new page.</p> <p>Select a specific filtering scheme from the list if you want the visualization to always use that filtering scheme. Moving the visualization to another page with a different filtering scheme will not affect this setting.</p> <p>If you choose to limit data using more than one filtering scheme, the intersection of the filterings will be used. This means that the visualization will show only the data that is kept visible by all of the filtering schemes.</p>
<b>Expression</b>	<p>You can limit what data should be available for a certain visualization using an expression. If you have defined a limiting expression, it will be shown in the <b>Data limiting</b> list.</p>
<b>Streaming data time range</b>	<p>This option is available if the visualization is configured with a data table containing streaming data.</p> <p>Select this option to show data from a limited time window.</p> <p>The time range could be very short, for example containing data from the last 5 seconds, or much longer, for example containing data from the last week or month.</p> <div data-bbox="908 1514 951 1562" data-label="Image"></div> <p>You can specify a time range directly on a time axis in a visualization. Just click on the axis to open a popup menu.</p> <div data-bbox="1018 1457 1508 1717" data-label="Image"></div>

## Result

The visualization is updated according to your configurations.

To remove an added **Data limiting** from a visualization, in the visualization properties, click on the x next to the limiting you want to remove in the list. See [Making a column not respond to data limiting](#) on page 318 to remove data limiting only from one column in a graphical table.



## Using markings for data limiting

A visualization can be configured to show only data that has been marked in another visualization or visualizations, or even show only data that has been marked within the visualization itself. That is, the configured visualization will be based on a limited amount of data, determined by what is currently marked in the limiting visualizations.

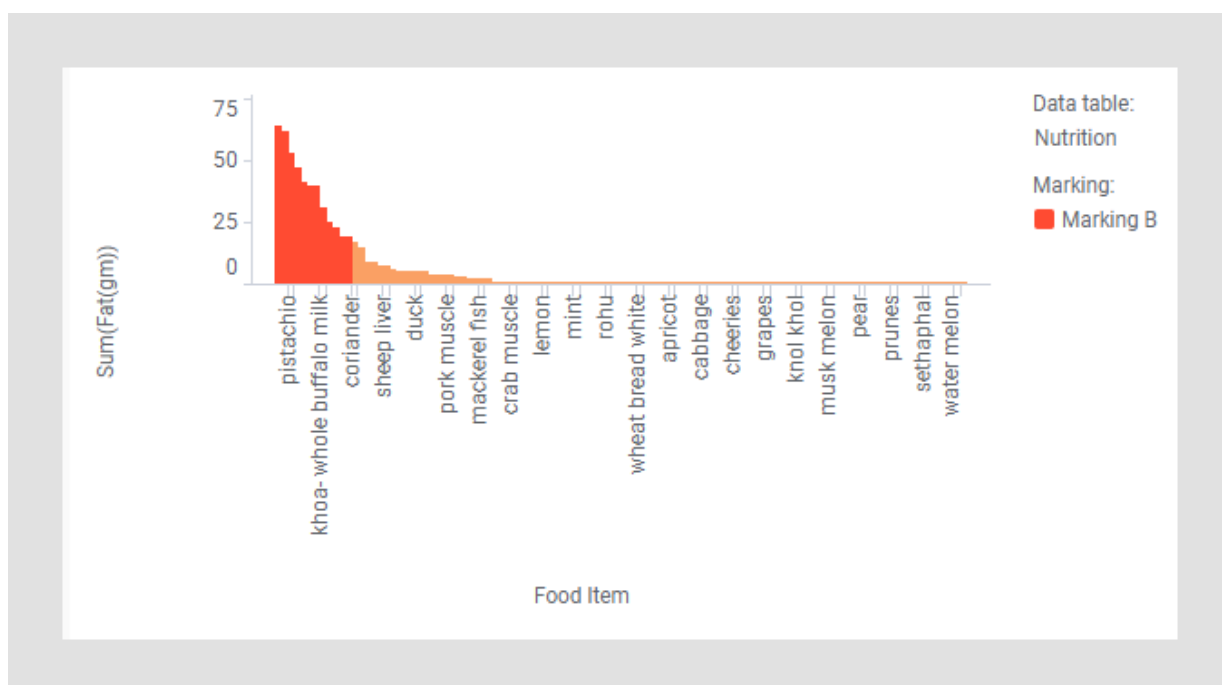
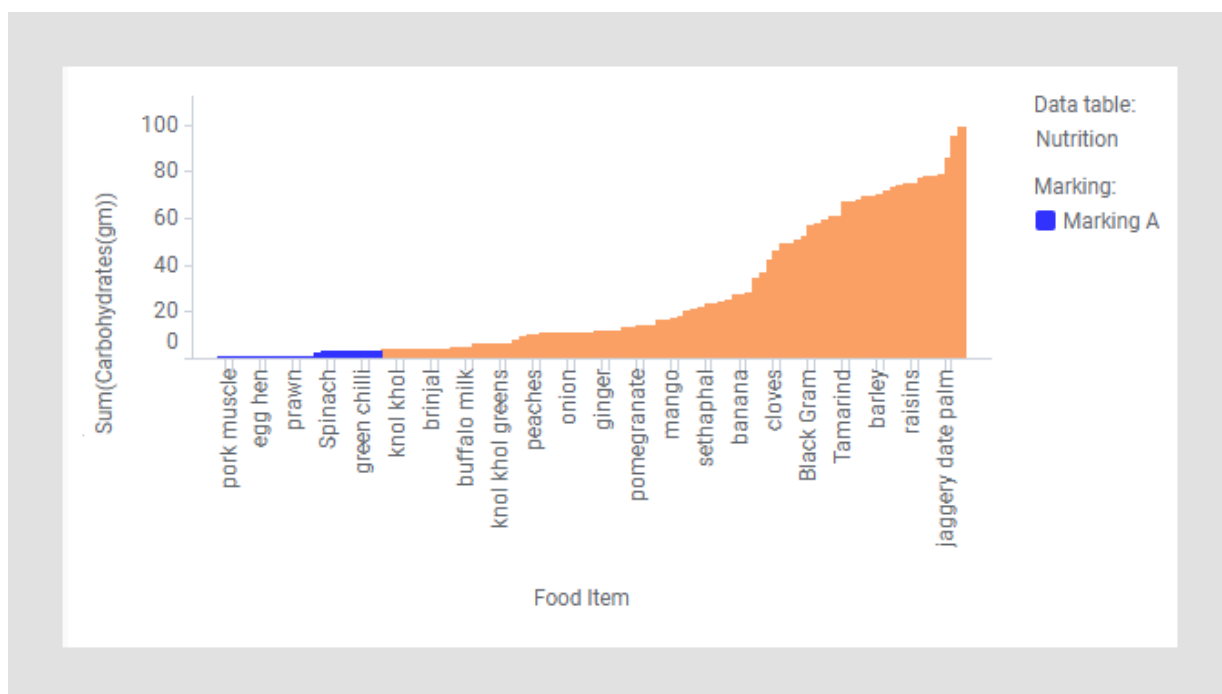
The most common use case of limiting data by markings is drilling down into [details visualizations](#). This example shows an additional use case. It illustrates how the limited data can be retrieved from different visualizations that use different markings. When more than one marking is used to mark data in the limiting visualizations, you can combine the marked data from the different markings in two ways; an intersection (AND), or as a union (OR). The settings is available in the **Data** section of the visualization properties.

The example data table contains nutrition data:

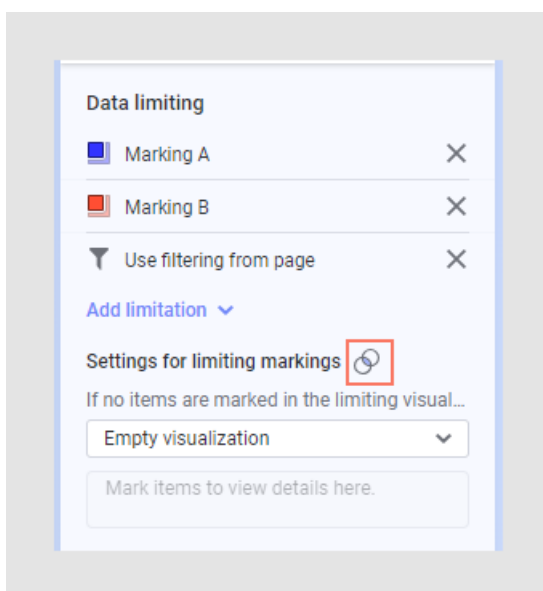
Food Item	Fat(gm)	Carbohydrates(gm)	Calorie(Ecals)	Protein(gm)	Minerals(gm)	Fibre(gm)
Black Gram Whole	5	61	360	17	3	4
Black Gram Roasted	5	58	369	22	2	1
Soyabean	19	21	432	43	5	4
Lentil	1	59	343	25	2	1
Red Kidney bean	1	61	346	23	3	5
Black Gram	0	57	321	22	3	5
wheat Bread Brown	1	49	244	9	0	1
wheat bread white	1	52	245	8	0	0
wheat Flour whole	2	69	341	12	3	2
wheat flour Refined	1	74	348	11	1	0
rice raw milled	0	78	345	7	1	0
rice flakes	1	77	346	7	2	1

### Limit data to intersection of marked items

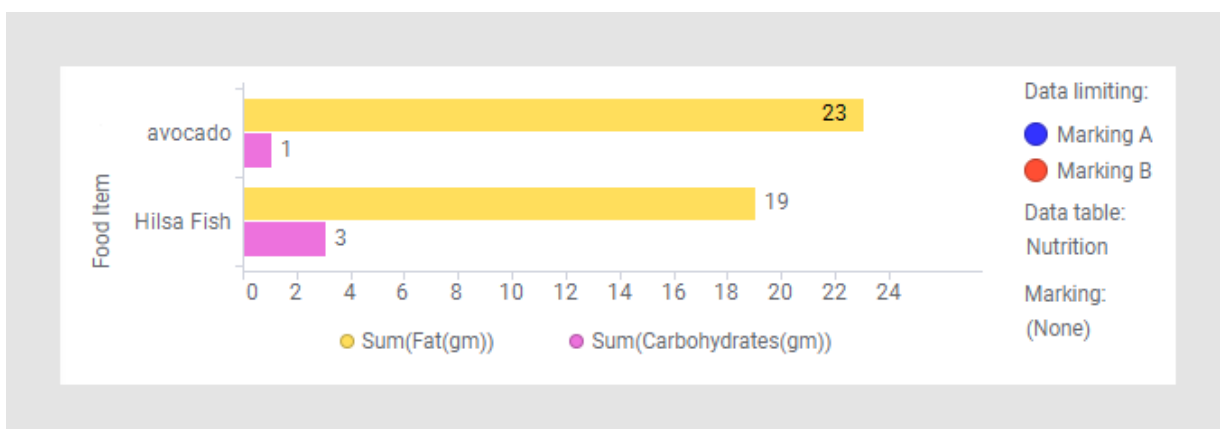
In a type of diet it is recommended to eat food with low carbohydrate values and high fat values. To identify these food items, the bar charts below are created. In the top bar chart, food items with the lowest carbohydrates values are marked using Marking A, and in the bottom bar chart, food items with the highest fat values are marked with Marking B.



Assume you wish to identify food items with low carbohydrate and high fat values. Simply create another bar chart, and make the settings shown below in its **Data** settings.

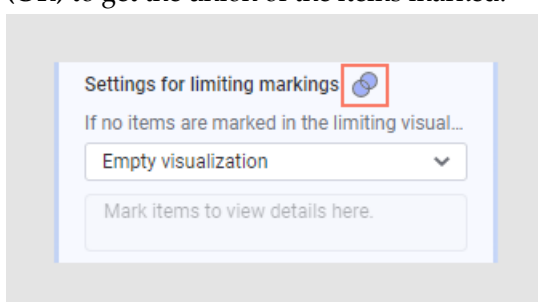


That is, make sure both Marking A and Marking B are selected under **Data limiting**, and, next to **Settings for limiting markings**, choose the **All markings (AND)** option to get the intersection of the items marked in the two bar charts. Among all food items, the items below are the ones that fulfill both the criteria:

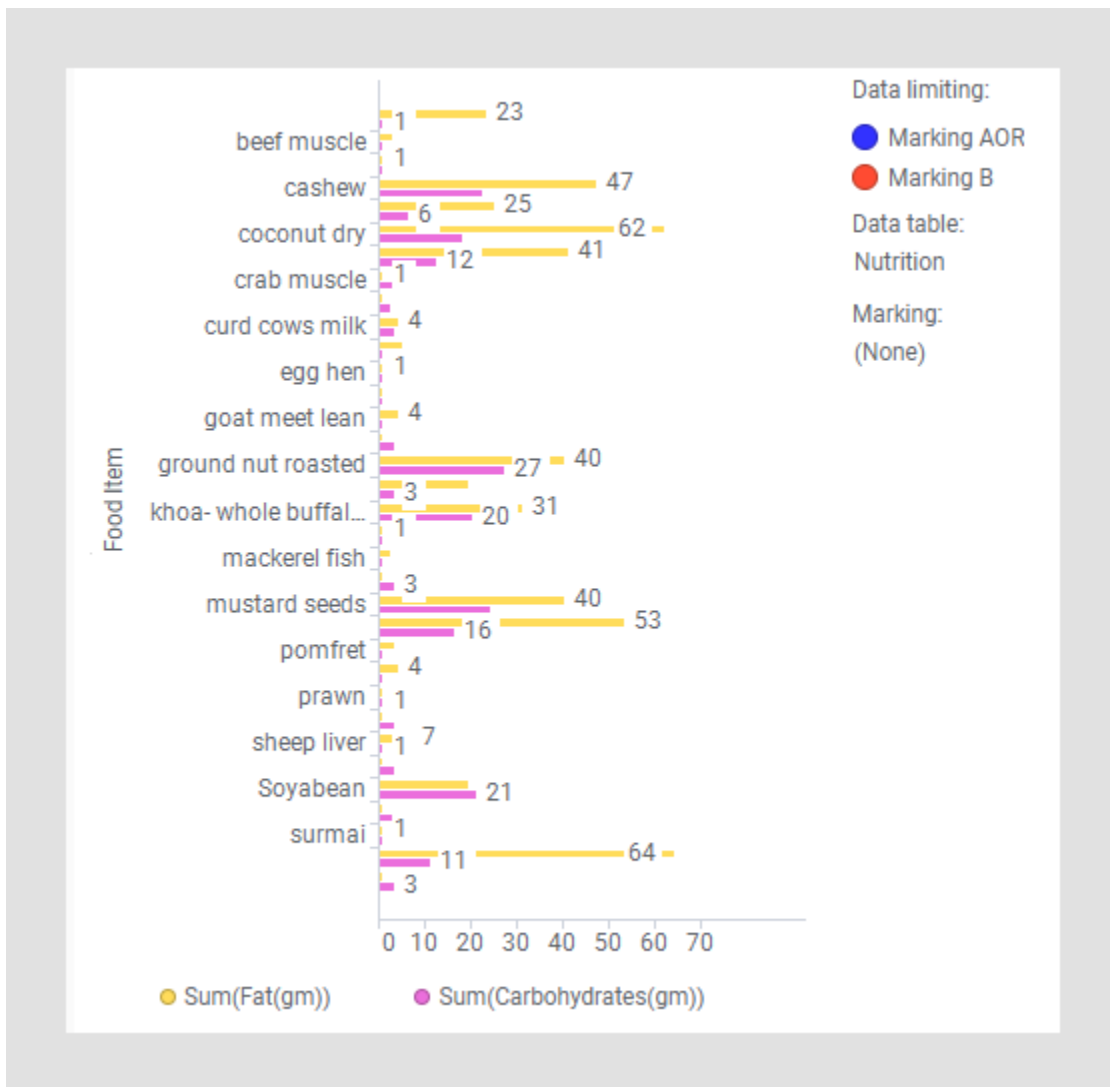


### Limit data to union of marked items

If you are instead interested in identifying food items fulfilling at least one of the criteria, that is, items with either low carbohydrate or high fat values, change **Rows must be included in** to **Any marking (OR)** to get the union of the items marked:



The items below are the ones that fulfill at least one of the criteria:



## Limiting the data in a visualization using filterings

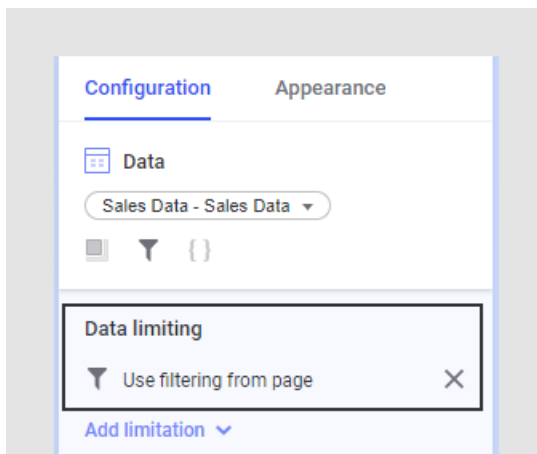
The data limiting settings of a visualization let you define how different filtering schemes in the analysis should affect the visualization. Using data limiting, you can make visualizations on a single page respond to different filtering schemes, or not be affected by filtering at all.

### Prerequisites

Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

By default, filter settings are the same in the entire analysis. If you want to apply various filter settings on different pages, filtering schemes can be created. Which filtering scheme to use on a page is selected at the top of the **Filters** panel (installed client only).

The **Use the current filtering from the page** limitation is added by default when creating a new visualization, and it makes the visualization always respond to the filtering scheme that is used on the page where the visualization is located. If you move the visualization to a new page, then the visualization will automatically start reacting on the filtering scheme that is used on the new page.



### To remove all filtering from a visualization

You can configure a visualization not to respond to any filtering at all. For example, if you want one visualization on a page to always show all the available data.

#### Procedure

1. Click to select the visualization to configure.
2. Open the visualization properties and locate the **Data** section.
3. Click to expand the **Data** card and, under **Data limiting**, click the **X** next to **Use the current filtering from the page** (and the **X** next to any other filtering limitation that might have been added).  
The visualization is no longer limited by any filtering.

### To make a visualization respond to a specific filtering scheme

You can select a specific filtering scheme from the list if you want the visualization to always respond to the filtering done using that filtering scheme. Moving the visualization to another page with a different filtering scheme will not affect this setting; the visualization remains tied to the specified filtering scheme.

#### Procedure

1. Click to select the visualization to limit.
2. Open the visualization properties and locate the **Data** section.
3. You probably want to remove the default limiting by clicking the **X** next to **Use the current filtering from the page**, as described above.
4. Under **Data limiting**, click **Add limitation** and choose **Filtering**.
5. In the Filtering schemes dialog, click to select the filtering scheme of interest.



You can create a new filtering scheme if needed.

## To use more than one filtering scheme for data limiting

If you select to limit data using more than one filtering scheme, the intersection of the filterings will be used. This means that the visualization will show only the data that is made visible by all of the specified filtering schemes.

Add more filtering scheme limitations as described in the steps above.

## Limiting the data in a visualization using an expression

You can limit what data should be available for a certain visualization using an expression. This way, you can set a "hard filter" on a visualization to make sure that certain values are never included in the calculations behind the visualization, regardless of any filtering in the analysis.

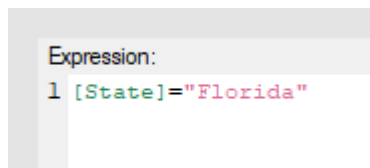
### Prerequisites

Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

For example, you can make sure that one visualization only shows data for a certain state by adding the expression similar to `[State]="Florida"`. If you combine this expression limiting with the usual filtering from page, you will see data for Florida, but only for the rows currently filtered to. If you remove all filtering limiting, you will see the results for all data that matches the expression. For other examples of data limiting, see [Adding data limitations for a visualization](#) on page 555.

### Procedure

1. Click to select the visualization to limit.
2. Open the visualization properties and locate the **Data** section.
3. Under **Data limiting**, click **Add limitation** and choose **Expression**.
4. In the opened dialog, enter an expression in the **Expression** field. In this example, we use `[State]="Florida"`. You could also use this limitation together with a custom property and a property control in a text area to be able to switch state. See [Limiting data using a property expression](#) on page 256 for more information.



In the expression field, you can insert columns and functions, or enter text as in any standard text editor. In this context, the expression must represent some type of condition which can be either true or false (a boolean expression), because it will be used as a filter.



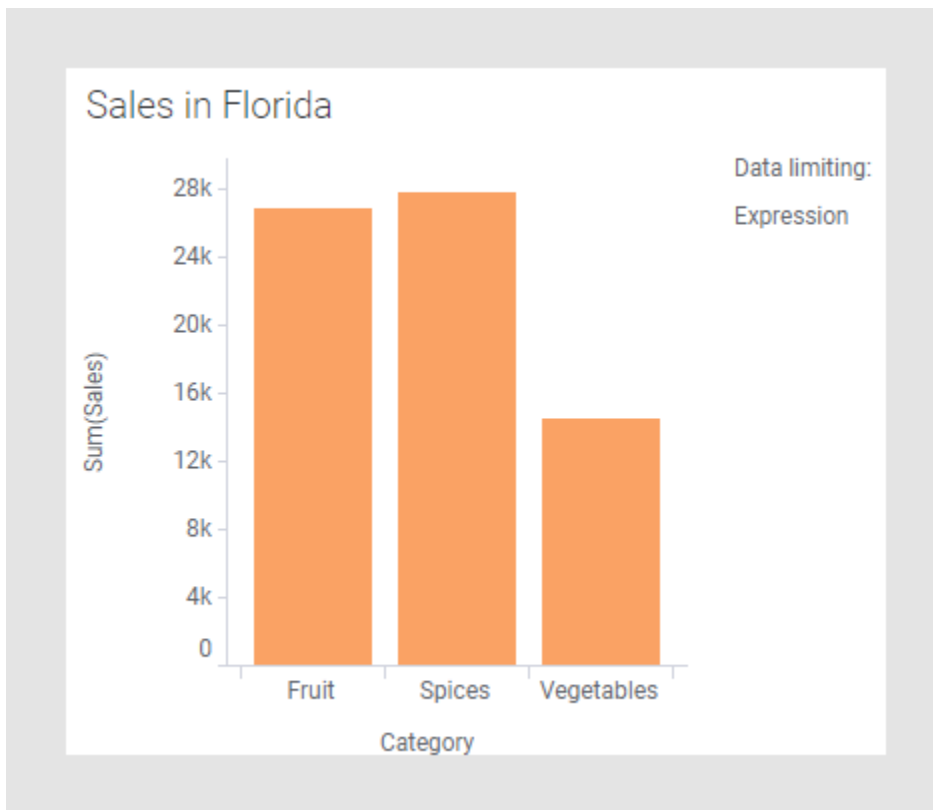
To edit an existing expression, click the pencil icon next to the expression and make your changes in the dialog.



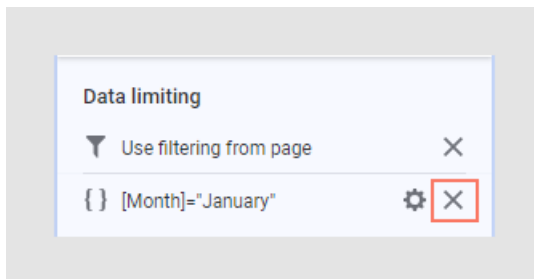
When using the installed client to specify the expression, you might find the **Resulting expression** field useful, if preprocessor functions (such as `${PropertyName}`) are used in the expression. It shows the expression after all occurrences of the property have been replaced with its current value or values. Read more in [Using document, data table or column properties in an analysis](#) on page 250.

### Result

The visualization is limited to only show data that fulfils the conditions of the expression.



To remove an added limiting expression from a visualization, in the visualization properties **Data limiting** section, click on the **X** next to the expression in the list.



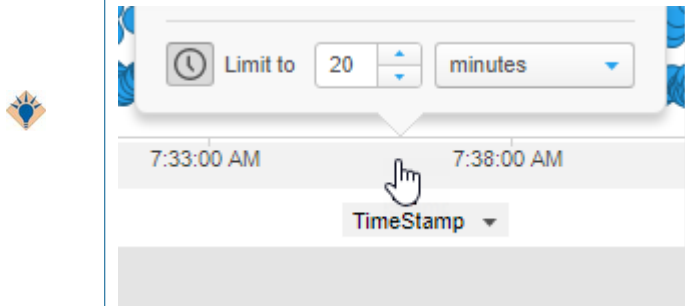
## Adding a streaming time range to a visualization

When you are working with streaming data and you want to limit the data in the visualization to a certain time range, you can add a **Streaming time range** limitation. This way, you can choose to only show data from the latest 5 minutes (or similar).

For more information about streaming data, see [Working with streaming data](#) on page 52.



You can specify a time range directly on a time axis in a visualization. Just click on the axis to open a popup menu.



For more options to specify the time range, use the **Visualization properties**.

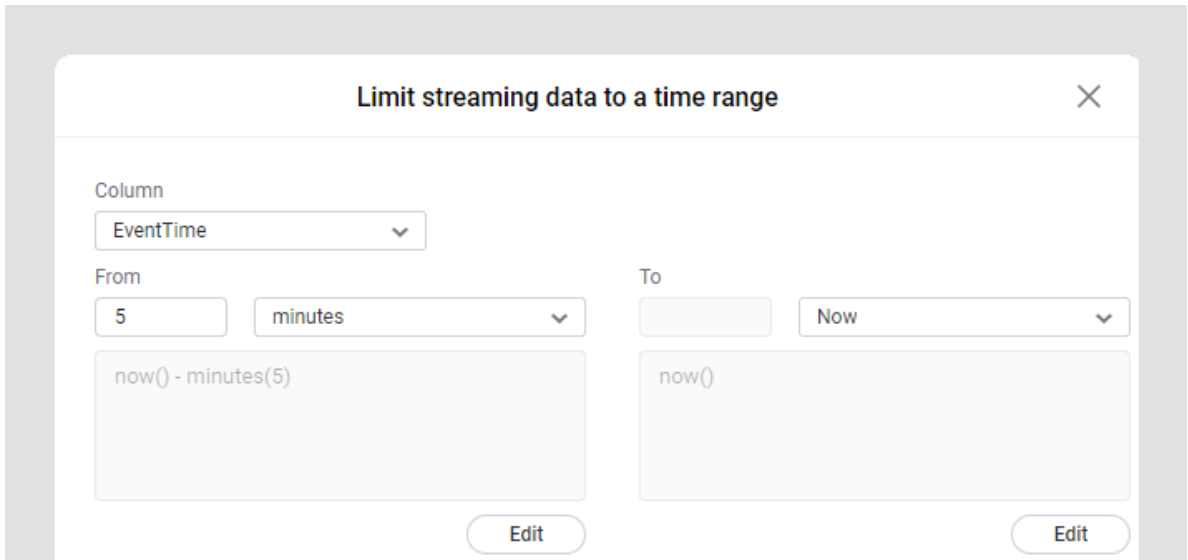
### Procedure

1. Click to select the visualization to configure.
2. Open the visualization properties and locate the **Data** section.
3. Under **Data limiting**, click **Add limitation** and choose **Streaming data time range**.



You can only use one streaming data time range in each visualization, but you can click the settings icon to edit a previously added limitation.

4. In the Limit streaming data to a time range dialog, configure the time range as desired.



Option	Description
<b>Column</b>	Select the <code>DateTime</code> column you want to use when specifying a time range.
<b>From</b>	Specify a starting point for the time range by selecting a number and a time unit.
<b>To</b>	Specify an end point for the time range by selecting a number and a time unit.
<b>Edit</b>	Click <b>Edit</b> to open the Limit Data Using Expression dialog (installed client), or the Custom expression dialog (in web clients), where you can write your own expression to specify a starting point or an end point for the time range.

- When you are done, click **OK**.



When you select multiple visualizations and open the Limit streaming data to a time range dialog from the Visualization properties panel, a streaming data time range will be added to all selected visualizations. Verify the settings before you apply them by clicking **OK**.

## Renderer settings

If the data in a column can be shown as something other than text, like a link, an image, or a shape, you can configure this using the renderer settings in the visualization properties. The renderer settings is available in some visualizations only, for example in tables, but might also be used in some labels or tooltips.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.



If you apply renderer settings from the visualization properties, the settings are applied to the selected visualizations only. If you apply renderer settings from the Add/Edit Default Renderer dialog in **Tools > Options** in the installed client, the settings are applied to all new visualizations.

### Image from URL renderer settings

Your Spotfire Administrator might have specified that only certain URIs should be allowed to link to. If that is the case, then a list of the **Allowed URLs** will be shown in the dialog. An administrator can add more addresses to the **Whitelist for Allowed URIs** preference found under **Application Preferences** in **Tools > Administration Manager**, in the installed client.

If an allowed URI includes a wildcard character (for example, `http://*.example.com/`), then you can use any path within that specified domain.

To make a directory on a company network location allowed, specify the path to the folder in the preference. The path must start with `file://`.

For example:

```
file:///Q:/MyCompany/Images
```



Because local network paths may be unavailable to other users, the recommended procedure is to use a URL rather than a local network path when publishing analyses to the library.

For users running Spotfire without a server, see [Updating the whitelist for allowed URIs when running Spotfire Analyst without a server](#) on page 567.

### Width and Height

If the website allows, you can also specify the width and height of the rendered image (in pixels):

```
http://MyCompany.com/Images/{$/image?width={$width}&height={$height}
```

If `$width` and `$height` are included in the link template, the image will be rendered using the width and height specified by your current table row height/column width, or, in the case of labels or tooltips, by the size set using the size slider. If not included, the default size will be rendered and changing the size will simply enlarge or shrink the default image.

### Geometry renderer settings

If your data contains geometries, you can sometimes show the geometric shapes in the visualization.

In the visualization properties, you can also specify the **Fill color**, **Border color**, and the **Border weight** (thickness of the line) of the geometry.

## Link renderer settings

This renderer is used to show the values in a column as clickable links.

Example:

If the link template is entered as follows, `http://www.domain.com/{$.html}`, the resulting link would redirect you according to the table below:

Cell value	URL
a	<a href="http://www.domain.com/a.html">http://www.domain.com/a.html</a>
b	<a href="http://www.domain.com/b.html">http://www.domain.com/b.html</a>
c	<a href="http://www.domain.com/c.html">http://www.domain.com/c.html</a>
d	<a href="http://www.domain.com/d.html">http://www.domain.com/d.html</a>
e	<a href="http://www.domain.com/e.html">http://www.domain.com/e.html</a>



If your company wants to link to something other than http-addresses, an administrator has the possibility to modify the preferences **Allow Additional Uri Schemes** and **Additional Uri Schemes** in the installed client using Administration Manager.

## Updating the whitelist for allowed URIs when running Spotfire Analyst without a server

When running Spotfire without a server, the whitelist will be enabled by default, and it will be empty from the start. All URIs in all analyses using **Image from URL** and **Web Page Panel** must be added to the whitelist, if you want to use these features.

### Prerequisites

You must have access to the installed client.

### Procedure

1. Make a comma-separated list of all URLs you want to be able to use with **Image from URL** or **Web Page Panel**.
2. Close Spotfire.
3. Go to AppData on your computer and, in the Modules folder, open the Spotfire DXP Forms folder. For example: `C:\Users\{user name}\AppData\Local\Spotfire\{version number}\Modules\Spotfire DXP Forms_62.0.18509.4618`  
Note that the Modules folder is hidden by default.
4. Open `Spotfire.Dxp.Main.dll.config` in a text editor (for example Notepad).
5. Near the bottom of the file, locate the setting `Desktop_Preference_WhitelistForSecuritySensitiveUris`.
6. As value, insert the comma-separated list of URIs created in step 1. If desired, you can add a wildcard (\*) to the first part of the host-name for the URI, to include all URIs within a certain domain. The final result could look something like this:

```
[....]
<!--
```

A whitelist of comma-separated uris that are allowed to be used in certain situations.

This setting is only used if Desktop\_Preference\_UseWhitelistForSecuritySensitiveUris is set to true.

-->

```
<setting name="Desktop_Preference_WhitelistForSecuritySensitiveUris"
serializeAs="String">
```

```
<value>http://example.com/some/very/special/deep/path/,http://
*.example2.com/,http://*.example3.com/some/path/</value>
```

```
</setting>
```

```
</Spotfire.Dxp.Application.Properties.Settings>
```

```
</applicationSettings>
```

```
<startup><supportedRuntime version="v4.0"
```

```
sku=".NETFramework,Version=v4.5"/></startup>
```

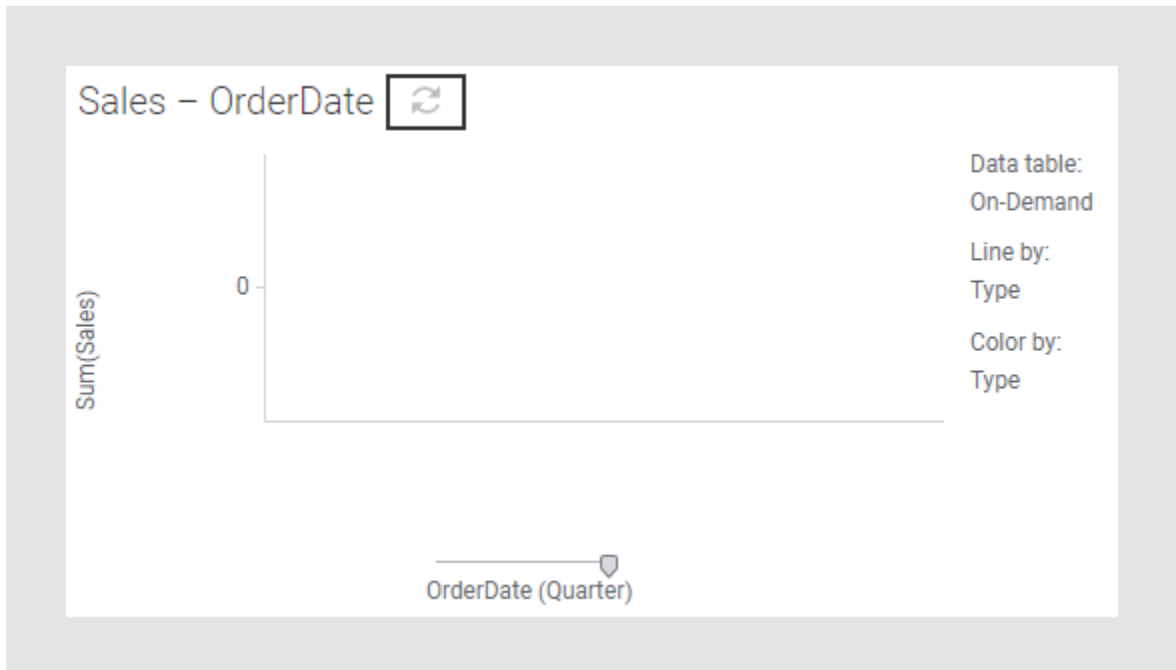
```
</configuration>
```

7. Save the file.
8. Start Spotfire.

## Updating a visualization that needs a manual refresh

If a visualization is based on an on-demand data table not configured to load data automatically, or if a visualization is based on a calculated data table (for example, visualizations created using the data relationships tool in the installed client) you must manually update the visualization when the input has changed.

When the input controlling the [on-demand](#) or calculated data table is changed, a refresh button is shown on the title bar of the visualization.



### Prerequisites


Anyone who has access to the analysis can update the visualization, as long as the title bar is visible.

## Procedure

1. Locate the visualization to update.



If the title bar has been hidden, an author can open the visualization properties and select **Show title bar**.

2. Click the refresh icon .  
The visualization is updated.

## Filtering data

You can base the visualizations on the entire set of data, or you can base them on certain data values you have filtered to.

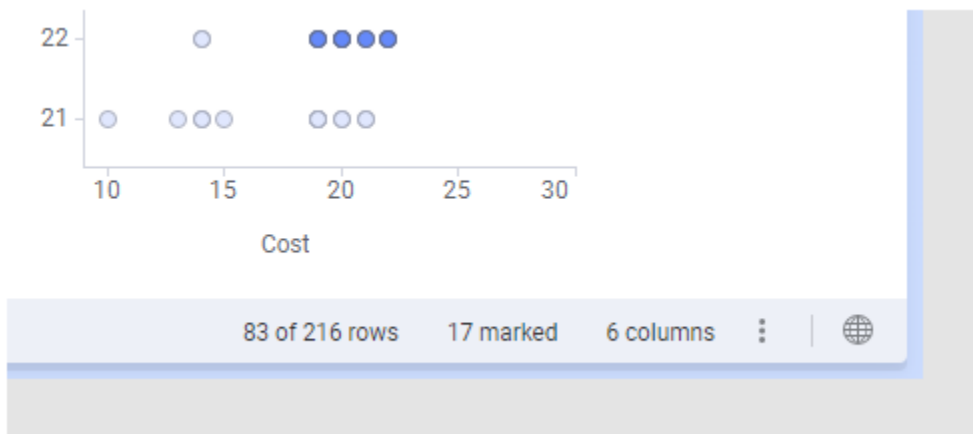
When you filter, certain data values in the data table are filtered out, and only the filtered values, that is the remaining values, are included in what is being visualized. All visualizations that are based on this data table will immediately update to reflect only the filtered values. However, you can anytime change what is filtered, or return to the full set of data.



Not only visualizations on the open page are affected; every visualization that is based on this data table will be updated in your analysis.

The filtering is handled from the **Data in analysis** flyout or the **Filters** panel. If you think of your loaded data as a table with rows and columns, each column gets a filter of its own in the panel. If any column value is filtered out, every row that contains that value will temporarily be taken away.

How many rows that remain after filtering, and the total number in the data table, can be viewed at the bottom of a page. In this example, the entire data table consists of 216 rows and 83 rows remain after filtering.



As data columns contain different types of data, their associated [filters are of different types](#) to make it easy to filter to the values you want to keep. You can however change the filter type that is suggested.

How to filter data is described in [Filtering data using the Data in analysis flyout](#) and [Filtering data using the Filters panel](#).

### Filtering data using the Data in analysis flyout

In your analysis, you can filter to data in which you are particularly interested and filter out the rest. The filtering can be handled from the **Data in analysis** flyout.

For more information, see [Filtering data](#).

#### Prerequisites

The **Data in analysis** flyout is open.

## Procedure

1. In the flyout, move the cursor over the column containing the values you want to filter out, and click **Show filter** .

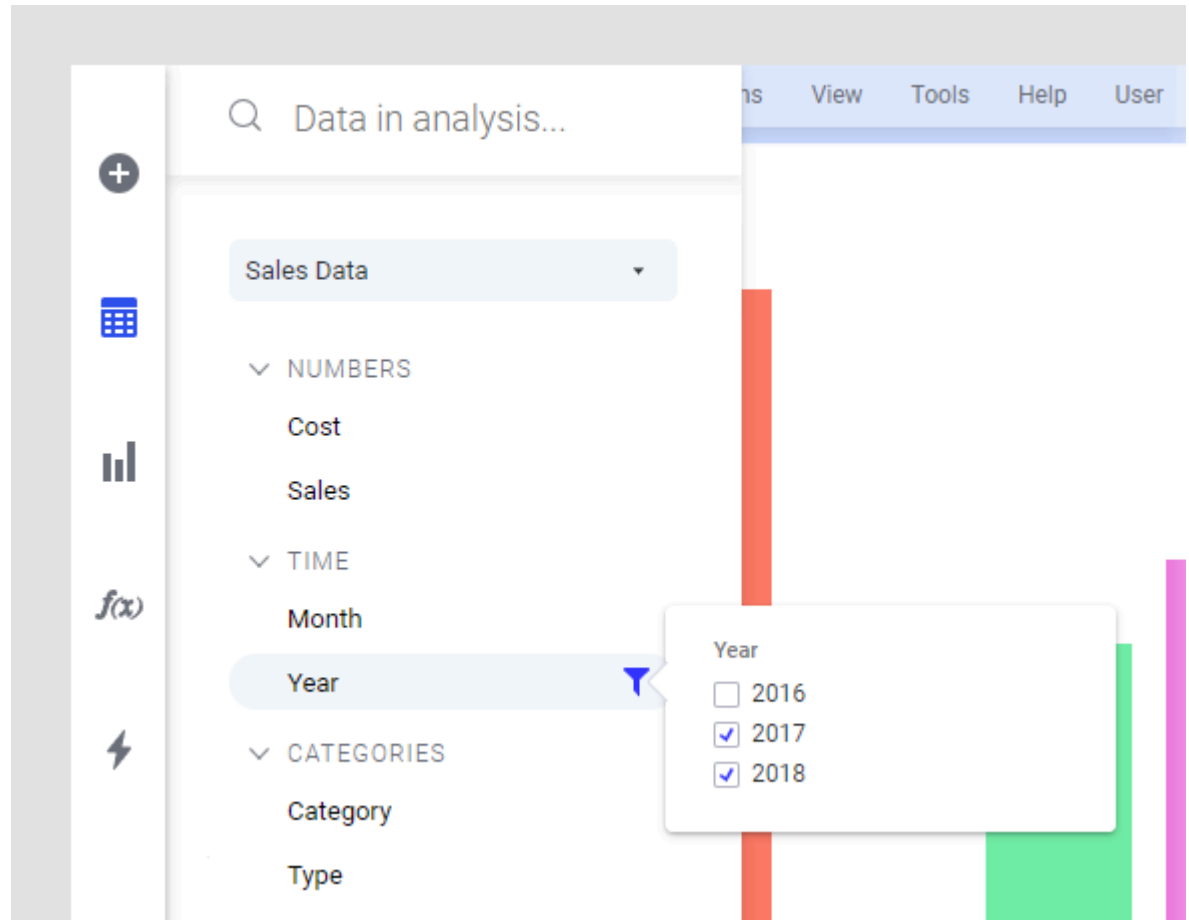


If you are working with in-database tables, filters might not have been created yet. Use the **Click to create a filter** button to create one. When working with cube data it is not possible to create filters for measures or sets, only for dimension columns. This is because the cube calculates the measures in the context of the selected dimensions.



Creating filters can take some time, depending on the database used.

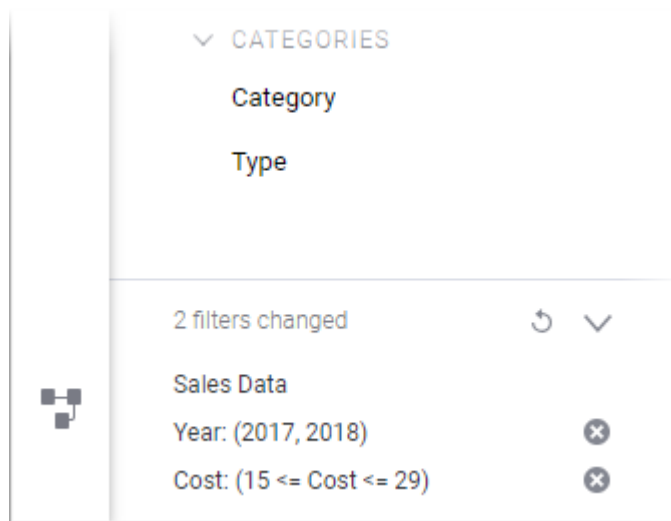
The filter associated with the column opens. One [type of filter](#) is shown below.




2. Use the opened filter to specify which column values to keep and which values to filter out.

3. Repeat from step 2 to filter data in another column.

Information about which columns have been filtered, and to which values, is available at the bottom of the flyout. If you want to modify the filtered data, you can open the filter from here by simply clicking it.



Click the  button to reset separate filters, and the  button to reset all filters.

To collapse the filter details, click the  button.



Any filtering action in the **Data in analysis** flyout also updates the filters in the **Filters** panel.

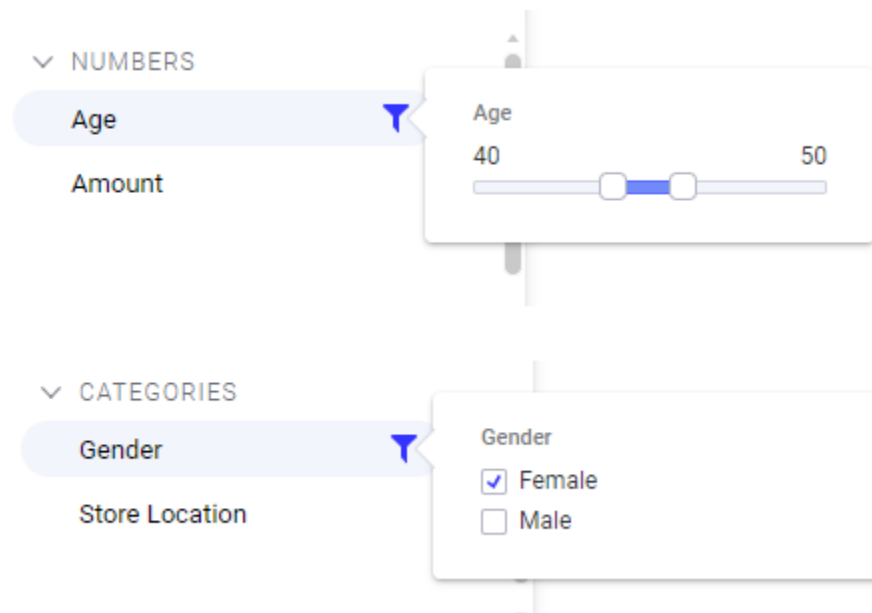


### Example

The data table below lists how much money each customer spent last time they visited a store. Also the customer's age and gender are registered.

Customer ID	Age	Gender	Amount
SSMM55001	61	Female	10013
SSMM55002	45	Female	6396
SSMM55003	55	Female	5586
SSMM55004	41	Male	885
SSMM55005	68	Female	4860
SSMM55006	56	Female	216
SSMM55007	74	Female	436
SSMM55008	77	Female	0
SSMM55009	62	Male	153
SSMM55010	65	Female	609
SSMM55011	54	Female	1384
SSMM55012	57	Male	260

What if you are interested in only showing purchases made by female customers between 40 and 50? Simply filter to these particular values using the **Data in analysis** flyout as shown below.



The data table, and every visualization based on it, will adjust to visualize only this limited part of the loaded data.

Customer ID	Age	Gender	Amount
SSMM55002	45	Female	6396
SSMM55016	48	Female	1730
SSMM55020	43	Female	2273
SSMM55023	44	Female	492
SSMM55028	42	Female	379
SSMM55037	48	Female	0
SSMM55042	44	Female	1139
SSMM55049	46	Female	835
SSMM55050	48	Female	1482
SSMM55051	41	Female	238
SSMM55060	49	Female	3086
SSMM55064	43	Female	0
SSMM55069	49	Female	1252
SSMM55070	47	Female	3510


## Filtering data using the Filters panel

In your analysis, you can filter to data in which you are particularly interested and filter out the rest. The filtering can be handled from the Filters panel.


For more information, see [Filtering data](#).

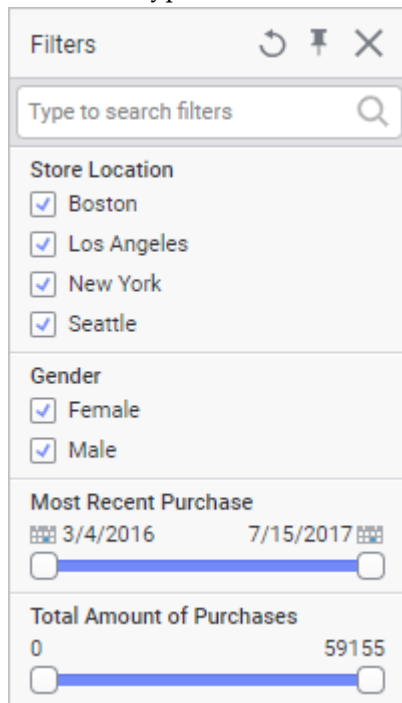
When you work with in-database data tables, the filters panel will initially be empty. If this is the case, you must create the filters you want to use.



To create a filter for a column in an in-database data table, move the cursor over the column in the **Data in analysis** flyout, and click **Show filter** , then **Click to create a filter**. When working with cube data it is not possible to create filters for measures or sets, only for dimension columns.

## Procedure

1. If the **Filters** panel is not visible, click **Filters**  on the menu bar.  
In the panel, you find filters associated with the columns in your data table. The example below shows two types of filters, check box filters and range filters.



**Filters**

Type to search filters

**Store Location**

☒ Boston

☒ Los Angeles

☒ New York

☒ Seattle

**Gender**

☒ Female

☒ Male

**Most Recent Purchase**

3/4/2016 7/15/2017

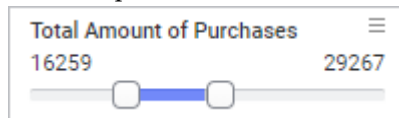
☐ ☐

**Total Amount of Purchases**

0 59155

☐ ☐

2. Locate the column containing the values you want to filter out.
3. Use its filter to specify which column values to keep and which values to filter out.  
An example of a filtered column is shown below.



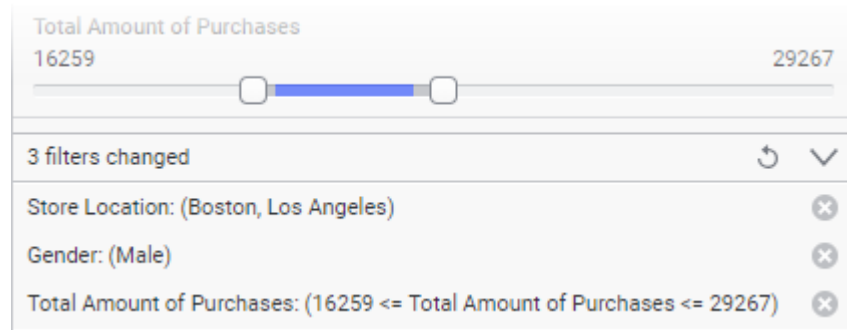
**Total Amount of Purchases**


16259 29267

☐ ☐

- Repeat from step 2 to filter data in another column.

Information about which columns have been filtered, and to which values, is available at the bottom of the **Filters** panel. If you want to modify what has been filtered in a filter, you can open the filter also from there by simply clicking it.



Above the expanded view is shown. To collapse the filter details, click the  button.

The filtering made can be reset. Click the  button to reset separate filters, or the  button to reset all filters.



Any filtering action in the **Filters** panel also updates the filters in the **Data in analysis** flyout.

## Filter types

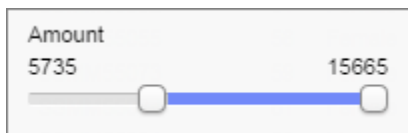
A number of filter types are available for filtering the loaded data to certain values in which you are particularly interested. Each filter type has its own characteristics, allowing you to select the most suitable way to filter your data.

How to handle the different filter types is described in

- [Using range filters](#)
- [Using item filters](#)
- [Using check box filters](#)
- [Using radio button filters](#)
- [Using text filters](#)
- [Using list box filters](#)
- [Using hierarchy filters](#)

## Using range filters

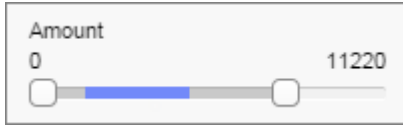
The range filter lets you filter to a range of values using a slider.



You can filter out values at both ends of the range, meaning that only rows with values within the chosen range remain in the visualization. Labels above the slider state the exact range.



A shade at either end of the slider indicates that currently there is no data visible in that part of the range because other filters in the analysis have filtered out those rows.




Double-click on the colored part to adjust the filter to only show the effective range.

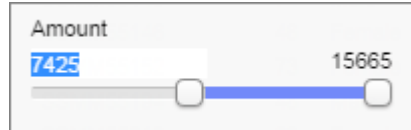
An important feature of the range filter is that the values are distributed on a linear scale according to the values of the data. Thus, if values are unevenly distributed, this will be reflected in the range filter. Note that this is not the case with item filters, where values are distributed at even intervals along the range of the slider, regardless of the actual numeric values.

### Procedure

1. Drag the left and right slider handles to change the lower and upper limits of the range.

If the range slider shows dates you can also click on the calendar icon  to set start and end dates.

2. Alternatively, double-click on a label, type the wanted end value, and press Enter.



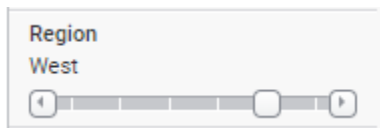
The labels show the lower and upper value of the range.

### Moving the selected range sideways

The currently selected interval of the range filter can be grabbed and moved to pan the selected range - this provides a powerful way of sweeping over different "slices" of a data table. Click and drag the blue portion of the modified range slider to do this. Observing the reactions of the other sliders to such a sweep can give some interesting clues to correlation between parameters in the data table.

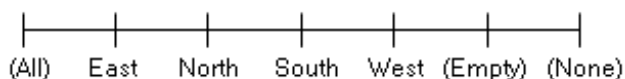
## Using item filters

The item filter is used to filter to a single value at a time using a slider.



On the slider, you select the value you want to keep, meaning that only rows with this value remain in the visualization. The slider makes it easy to switch between nearby items.

The item filter includes the (All) and (None) options. Selecting (All), the very first position to the left, means no filtering is made, and selecting (None) to the far right filters out everything. If the data column contains empty values, the (Empty) position will select these.



The label above the slider shows the selected item.



You can decide whether the (All) and (None) options should be visible. Right-click the filter, and use the **Include (All) as option** and **Include (None) as option** settings.

### Procedure

1. Drag the slider handle, or click the arrows at the edges of the slider, to select the single value you want to keep in the visualization.
2. Alternatively, double-click the label, type the value of interest, and press Enter.



## Using check box filters

In a check box filter, each unique value in the column is represented by a check box. It is used to filter to combinations of values in the data column.



Only rows containing values with selected check boxes remain in the visualization.

If there are empty values in the data column, there is also an (Empty) check box available.

Selecting all check boxes means no filtering is made, and clearing all check boxes means everything is filtered out.

Values that have been filtered out by other filters are indicated with gray text.



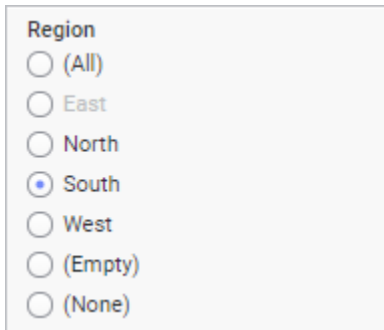
The check box filter cannot be used as filter type for columns containing more than 100 unique values.

### Procedure

- Select or clear one or several check boxes to determine on what values the visualization should be based.

## Using radio button filters

The radio button filter is used to filter to a single value.



A screenshot of a radio button filter interface. The title is "Region". Below it are seven radio buttons with the following labels: "(All)", "East", "North", "South", "West", "(Empty)", and "(None)". The "South" option is selected, indicated by a blue dot inside its radio button.

Only one value is selected at a time, meaning that only rows with this value remain in the visualization. When you create a new radio button filter, the (All) and (None) options are available in addition to the data column values. Selecting (All) means no filtering is made, and selecting (None) filters out everything. In case the data column contains empty values, also the (Empty) option is available.



You can decide whether the (All) and (None) options should be visible. Right-click the filter, and use the **Include (All) as option** and **Include (None) as option** settings.



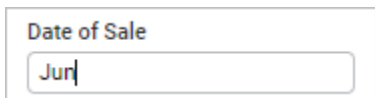
Values that have been filtered out by other filters are indicated with gray text.

### Procedure

- Click the radio button to select the value you want to keep in the visualization.

## Using text filters

The text filter is used to filter to values that match an entered text string.



A screenshot of a text filter interface. The title is "Date of Sale". Below it is a text input field containing the text "Jun".

You type a text string in the field. Only rows with values matching the text string remain in the visualization. Below you find tips on how to search for wanted values:

- The text filter is not case-sensitive.
- A **blank space** between two search words will result in rows containing both words.
- Search for **june OR may** to find rows containing either of those months.
- Use **\*son** to search for rows ending in "son" (Anderson, Jamesson, etc.), and use **\*sun\*** to search for rows containing the letters "sun" somewhere (Sunday, Asunder, etc.).
- Use quotation marks **"sample batch alpha"** to search for explicit phrases.
- Non-alphanumeric characters are not considered to be included in words, and if no quotation marks are used, matching is done on word boundaries. Add quotation marks to include special characters in the search string.

### Procedure

1. Type the text in the field.
2. Press Enter.

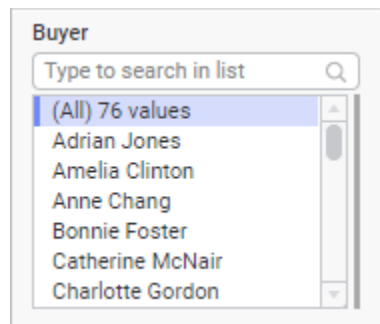
## Example

Assume you have a data column with all the months of the year, that is, January, February, March, ..., and December. Below you can see which values remain after typing different text strings.

Typed in Text Filter	Filtered rows
j	January, June, July
ju	June, July
jun	June
june	June

## Using list box filters

The list box filter is used to select a few values from a very long list of values present in the data column.



You select the values you want to keep, meaning that only rows with these values remain in the visualization. The list box filter also includes the (All) option. Selecting (All) means that no values are filtered out.



You can decide whether the (All) option should be visible. Right-click the filter, and use the **Include (All) as option** setting.

It is possible to search for values. Type a text string in the field above the list. Various search rules are described in [Using text filters](#).

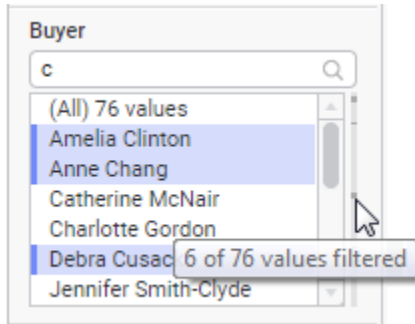
### Procedure

- Select values in the list by clicking them.
  - To select a consecutive group of values, click the first value, press Shift, and then click the last value.
  - To select non-consecutive values, press Ctrl, and then click each separate value.



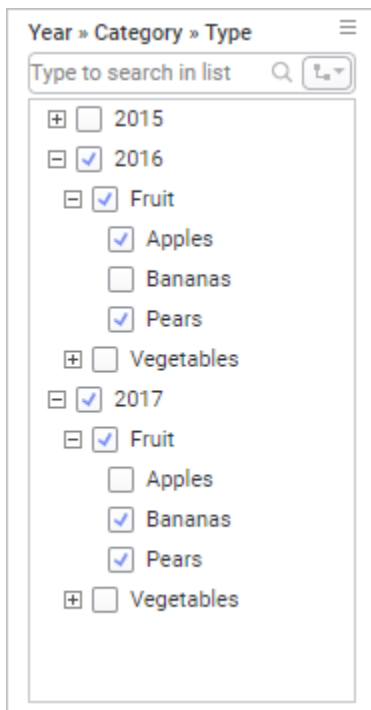
## Showing number of filtered values

The grey indicator on the right hand side of the filter shows where your selected values are located in the whole list. If you hover with the cursor over the indicator, a tooltip is displayed that shows how many values have been chosen in the list.



## Using hierarchy filters

A hierarchy filter represents the data in a hierarchy. The filter consists of an expandable tree view, where the categories within the different levels of the hierarchy are represented by check boxes.



Only rows containing values with selected check boxes remain in the visualization. For example, rows containing 2016, Fruit, and Apples are included in the visualization, but rows containing 2017, Fruit, and Apples are not included in the visualizations.



Values that have been filtered out by other filters are indicated with gray text.

Combining columns into a hierarchy filter will not automatically remove all other filters representing the same columns -- this must be done manually, if desired.



Creating a hierarchy with a very large number of nodes may take a long time. It may also result in a hierarchy filter with too many check boxes to be practically useful. Use the Filtering Scheme Properties or the Column Properties to edit the hierarchy and remove the column with too many unique values if this happens.

How to create a hierarchy filter is described in [Defining hierarchies](#).

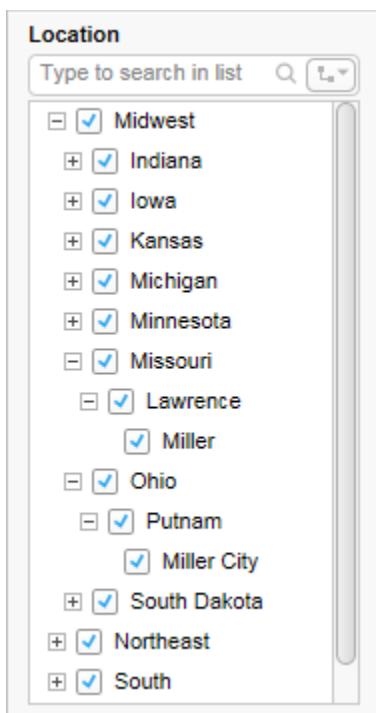
### Procedure

- Select or clear one or several check boxes to specify on which values the visualizations should be based.

### Search in hierarchy filters

The tree structure in a hierarchy filter can be extensive. To find specific parts in the hierarchy, you can type what you are looking for in a search field. Then you will find the items in the tree structure that begin with your typed text.

The search field is available at the top of the filter.



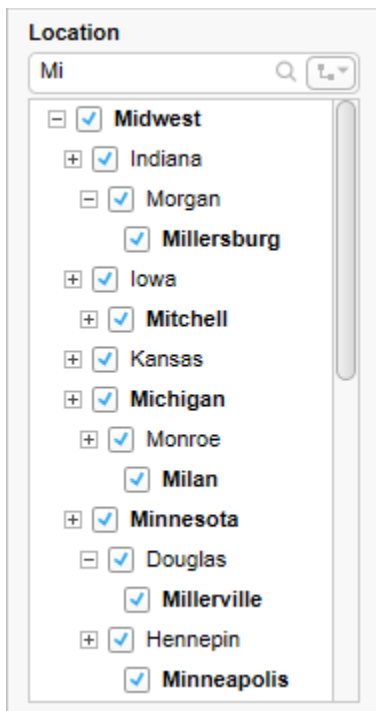
In the field, you can type words or parts of words, and also combine them with boolean operators. Each branch within the tree is searched, and branches found with tree items fulfilling the entire entered search criterion are returned. These items are shown in bold text.

Because searching the entire hierarchy can result in many matches, you can limit the search to a certain hierarchy level that you specify.

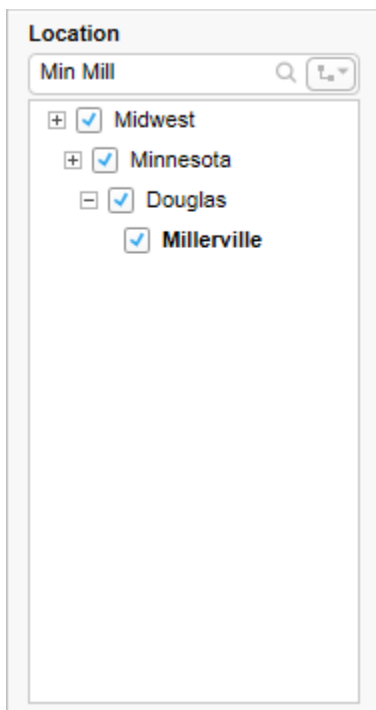
Different search examples follow.

### Typing the words or letters you are looking for

In the following example, the entire hierarchy is searched to find items beginning with Mi. Branches are returned that include an item fulfilling the condition, and the items are shown in bold text.



In the next example, the entire hierarchy is searched for branches where two conditions are fulfilled, that is, the returned branch must include both an item starting with Min and an item starting with Mill. A bold text item is the first item that fulfills the entire criterion in a found branch.

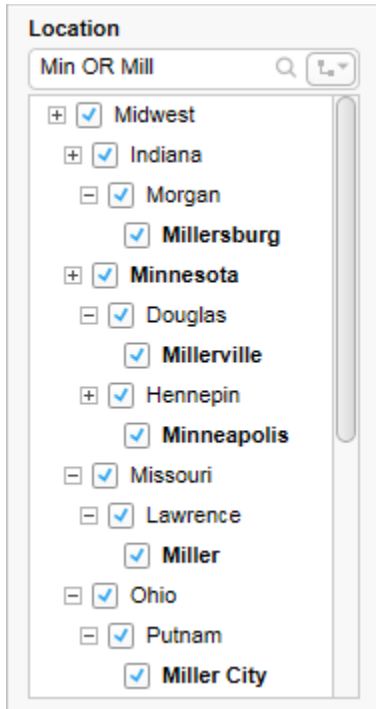


### Combining words with boolean operators

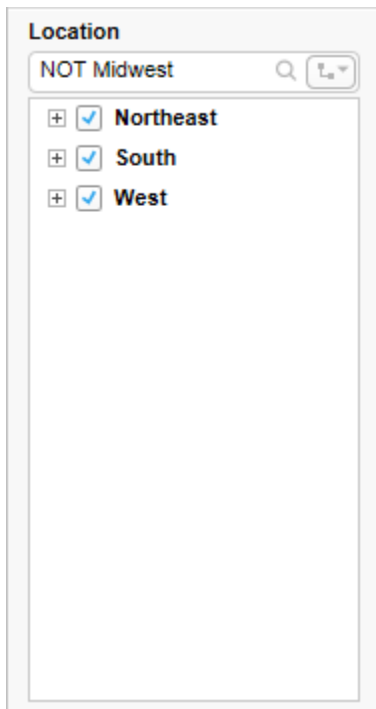
You can include the boolean operators AND, OR, or NOT in a search expression.

If you, in the search above, type Min AND Mill instead, the result will be the same, because typing two text strings in the field without a space is by default an AND search.

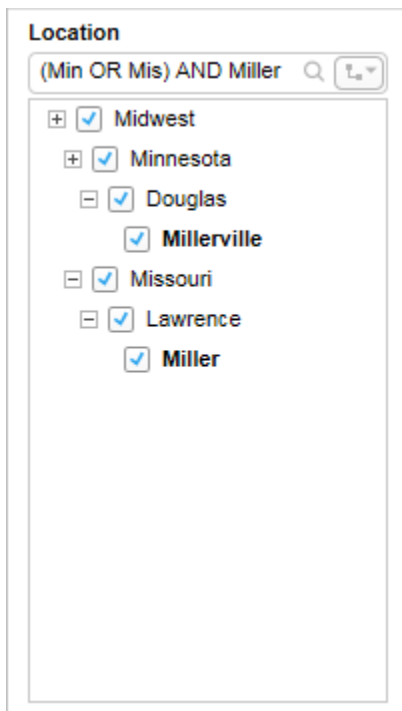
The result when typing Min OR Mill is shown below. All branches including items starting with Min or Mill are found. The matching items are in bold text.



The following example shows a search using the NOT operator. Every branch without items starting with Midwest is returned.

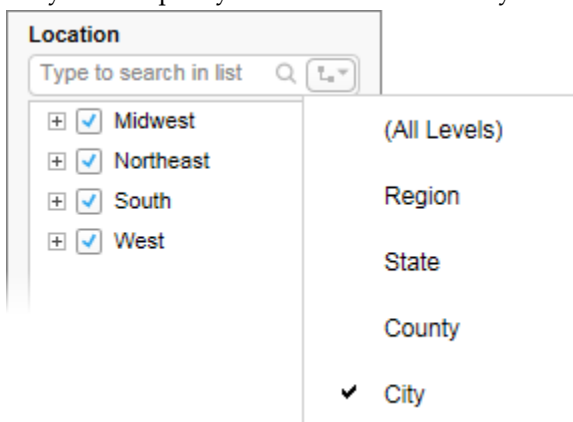


Boolean operators can be combined with parentheses in a search expression. For example, assume you want to locate items starting with Miller among a number of specific branches, say, branches starting with Min or Mis. Below, the expression is shown for this search.

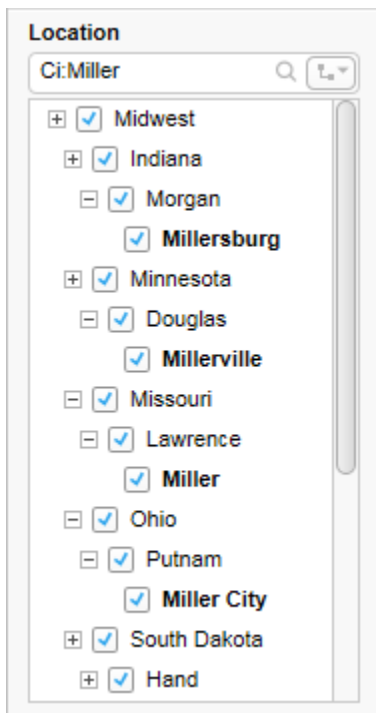


### Searching within a specific hierarchy level

You can narrow down a search using the drop-down list in the right part of the search field. From the list you can specify within which hierarchy level you want to perform the search.



Alternatively, you can type, preceding your search expression in the field, one or more of the initial letters in the level name followed by a colon as exemplified below. Ci: means the search is limited to the City level.




If you want to find cities named Miller exactly, you can type `Ci::Miller`, that is, type two colons. This expression will exclude Millersburg, Millerville, and Miller City above.

## Changing filter type


Data columns are by default associated with filters. The filters are of different types depending on the type of data. You can change the filter type though.

Each [filter type](#) has its own characteristics, and suits different needs.

### Prerequisites

The **Data in analysis** flyout is open. It is automatically shown when you create a new analysis. On the authoring bar, you can also click **Data in analysis** , when the analysis is in Editing mode.

### Procedure

1. In the flyout, move the cursor over the column in question, and click **Show filter** . The filter associated with the column opens.
2. Right-click the filter, select **Filter type**, and select the desired filter type in the opened list.

## Creating a filter transformation

Filtering out data using filters, or deleting marked rows, might sometimes not be permanent enough to remove unwanted data from your analysis, because reloading the data or resetting the filters might bring the undesired data back. If you want to make sure that values gets permanently removed, you can instead add a filter transformation.

A quick and easy way to create such a transformation is by filtering out the undesired data and creating a transformation based on the currently filtered rows.

## Prerequisites

You must have some data loaded in the analysis, and the analysis must be in **Editing** mode.

## Procedure

1. In the **Filters** panel, or in the **Data in analysis** flyout, filter your data to include only what you want to keep in the analysis.
2. When you are done, right-click the filter and select **Create filter transformation** from the pop-up menu.

In the Filters panel, you can press Ctrl and select more than one filter before right-clicking, if you want to base the transformation on more than one filter.

A transformation is added to the data table, where the filtered out data is excluded. You can remove the **Filter rows** transformation from the [Data canvas](#) at any time, or review the expression used on the **Information** tab. In the installed client, you can also edit the expression behind the transformation from the Data canvas.

## Result

The added transformation will exclude rows based on a boolean expression which is created from all the modified filters in the selection. Only those rows matching the expression (the rows for which the expression is True) will be included in the data table.

## Filtering schemes

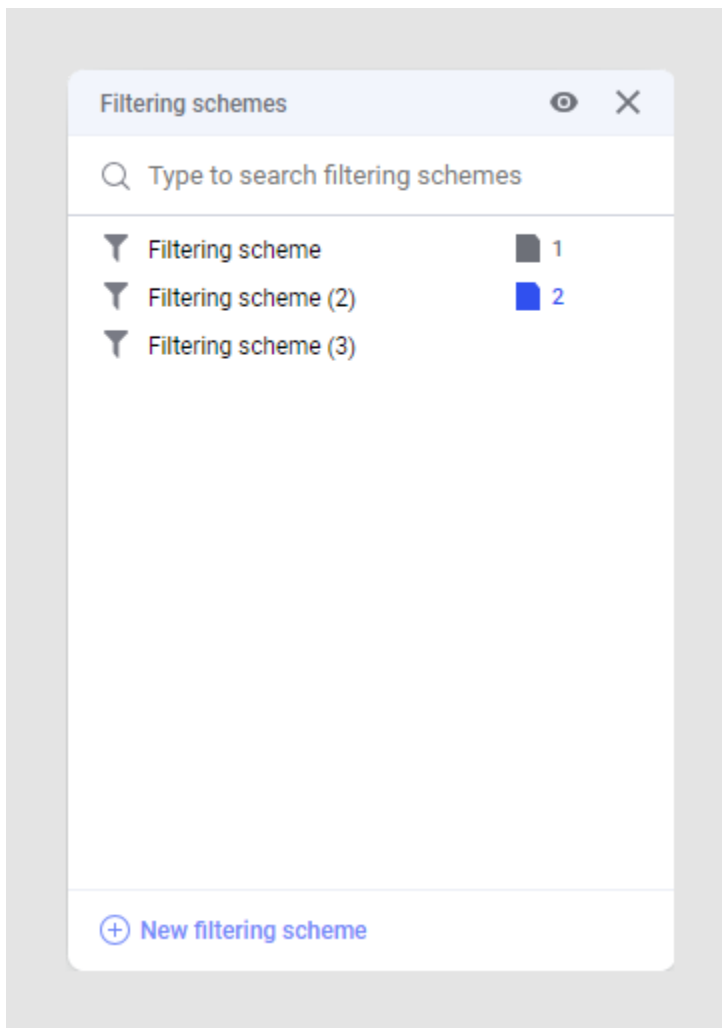
---

One of the main strengths of Spotfire is the ability it gives you to filter your data, that is, to control which data should be visible and used in calculations. This means that you can easily show or hide data for specific categories, change the time range to look at, step through a sequence of values one at a time, and so on. Filtering usually affects all pages and visualizations, but you can also define different filtering schemes to be used by different pages or visualizations.

As an author with the Advanced Filter Panel Properties license feature enabled, you have the possibility to add your own filtering schemes, which can be applied to the analysis per page or per visualization. This gives you the complete freedom to control which pages and visualizations will affect each other.

You can configure the filtering schemes to work on any combination of visualizations and/or pages in your analysis. For example, you can keep all the visualizations on all the pages related by using the same filtering scheme for all of them, or, you can choose to specify different filtering schemes for all the visualizations in an analysis, or any combination in between these two extremes. See [Limiting What is Shown in Visualizations](#) in the *Spotfire Analyst User Guide* to learn how to configure a visualization to use a different filtering scheme than the filtering scheme used on the page.

In the Filtering schemes panel (**View > Filtering schemes**), you can see on which page, or in which visualization, a certain filtering scheme is used.



## Adding a new filtering scheme

If you want filtering to be specific to just one or more pages or visualizations, rather than affecting the entire analysis, you can add one or more filtering schemes.

See also [Filtering schemes](#) on page 587 for general information.

### Prerequisites

You must be an author with the Advanced Filter Panel Properties license feature. You can only use the Filtering schemes panel when in **Editing** mode.

### Procedure

1. On the menu bar, select **View > Filtering schemes**.
2. In the Filtering schemes panel, click **New filtering scheme**.
3. Optional: Change the name of the filtering scheme by typing directly in the panel.

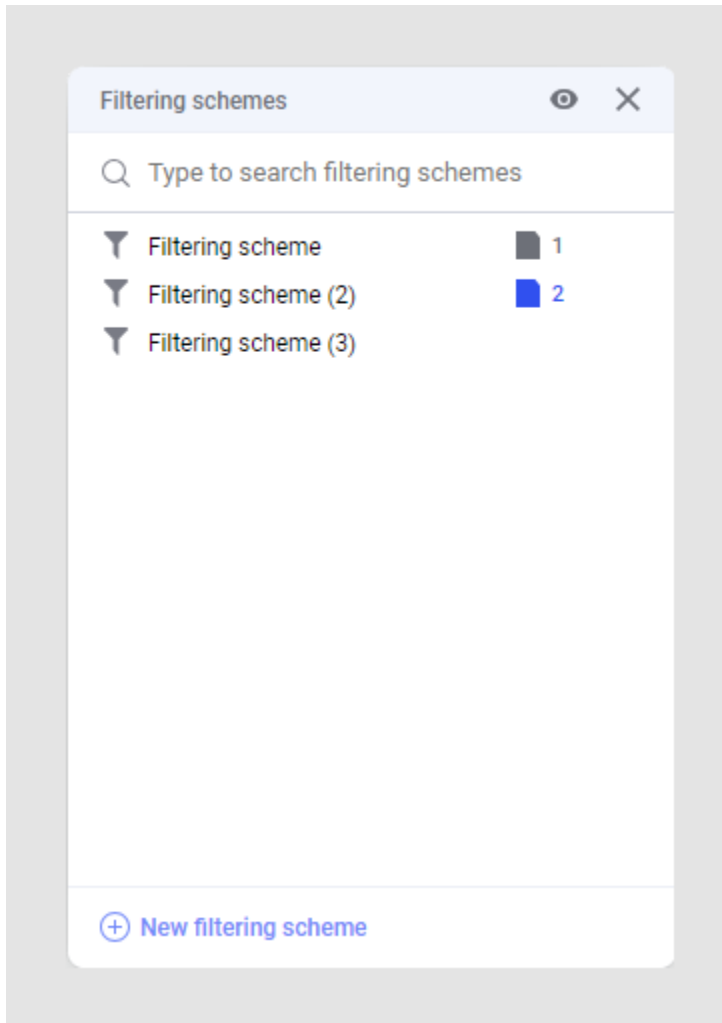


You can also open the Filtering schemes panel at a later time and select **Rename** from the context menu, or double-click on the name of a filtering scheme, to change its name.



## Result

The new filtering scheme is added to the list, and can be [used on a page](#), or as the limiting filtering scheme on a specific visualization.



## Changing the filtering scheme to use on a page


If you want to change the filtering scheme for a certain page in the analysis, you can do this from the menu at the top of the **Filters** panel. If the **Filering scheme** menu is hidden in the Filters panel, you can toggle the visibility of it from the Filtering schemes panel.

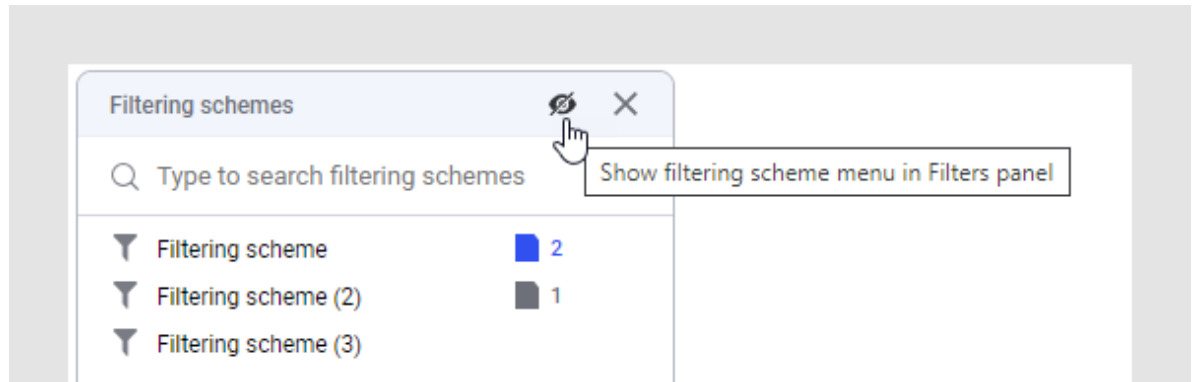
### Prerequisites

You must be an author with the Advanced Filter Panel Properties license feature. You can only use the Filtering schemes panel when in **Editing** mode.

See also [Filtering schemes](#) on page 587 for general information.

## Procedure

1. Make sure the **Filtering schemes** menu is visible at the top of the **Filters** panel. If it is hidden, select **View > Filtering schemes** and click on the icon **Show filtering scheme menu in the Filters panel** . Click the icon again to hide the menu.

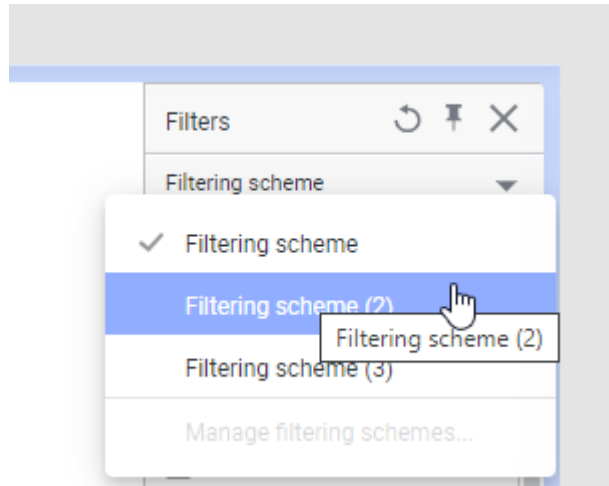


You must show the filtering scheme menu if you want to change the filtering scheme used on a page. However, when sending the finished document to a coworker for further analysis, you might want to hide the menu to save space in the filters panel and to reduce the risk of changing the filtering schemes by mistake.



Instead of changing the filtering for an entire page, you can use a different filtering scheme to limit a specific visualization from the **Data limiting** settings in the visualization properties.

2. In the **Filters** panel, select the desired filtering scheme.



Depending on what your filtering schemes are called, you might see different names in the list.

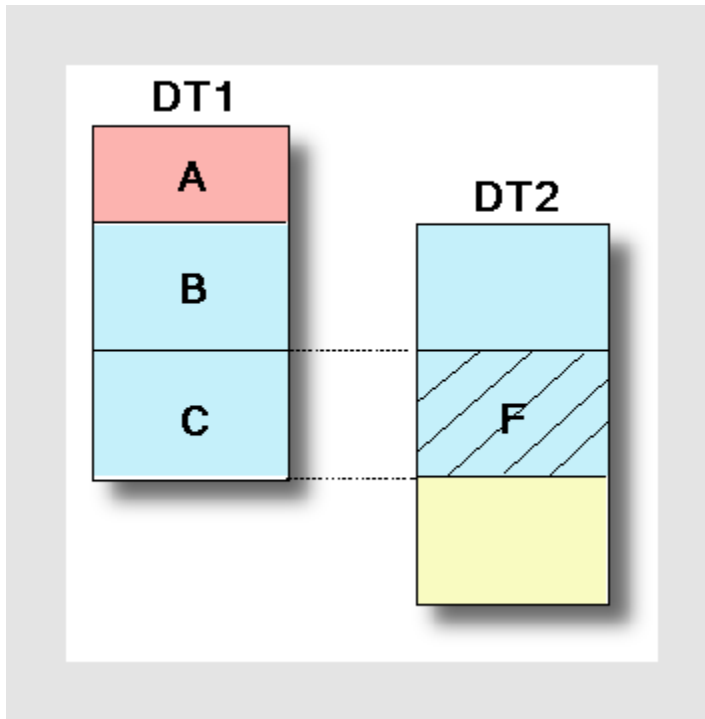
## Filtering in related data tables

When you have multiple data tables that are related to each other in your analysis, and the data tables do not include exactly the same values, you might want to handle filtering in the related data tables in different ways, depending on whether you are interested in the filtered values or the filtered out values in the columns that define the relation.

## Prerequisites

You must have more than one data table in the analysis and at least two of the data tables must have a relation configured. You can see all current relations and add, edit or remove relations from the **Relations** tab in the [Data tables overview](#).

To help show the difference between the three options available, here is an example with two related data tables, DT1 and DT2. Both DT1 and DT2 contain some values that are not available in the other data table (pink and yellow), but they also contain common values (blue):



A = Values in DT1 that are not available in DT2.

B = Values in DT1 that are available in DT2, but have been filtered out.

C = Values in DT1 that are available in DT2 and included in the currently filtered values of DT2.

F = The filtered values (values remaining after filtering) in DT2.

When the filtering management for DT2 is specified (from the DT1 data table header) the different options will give the following results:

### Include filtered values only

The first option will make all values that are only present in DT1 disappear from the visualizations using DT1, because only the values that are currently filtered in DT2 will be included. Hence, this option keeps only those values that are present in both data tables (and have not been filtered out).

In the example above, this means that only the values in C will remain after filtering in DT2.

### Exclude filtered out values

The second option will remove those values that have been filtered out from DT2 from all visualizations using DT1. Hence, this option keeps those values that are filtered in DT2 as well as the additional values from DT1.

In the example above, this means that A and C will remain after filtering in DT2.

### Ignore filtering (default)

The third option is to ignore any filtering done in the related data table completely. This way, all values that are available in the current data table will remain available.


In the example above, this means that A, B and C will all remain after filtering in DT2.



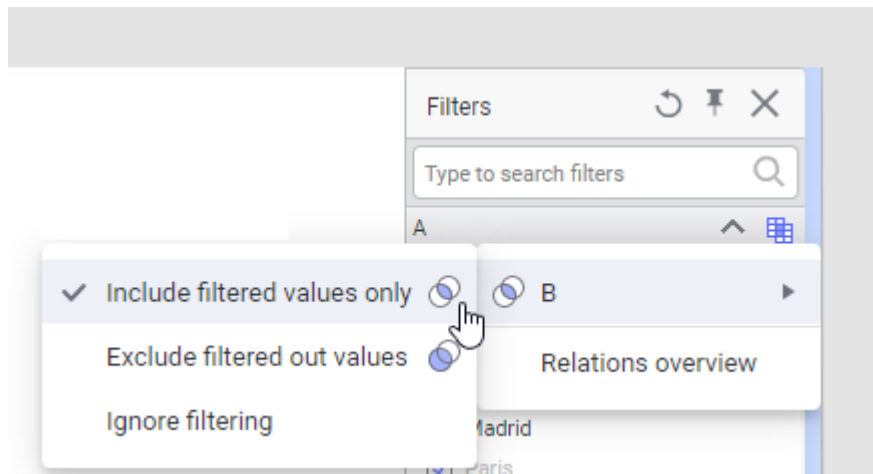
You must specify how each table should respond to filtering in all related tables separately, to be certain of what will be shown in the visualizations after filtering.

## Changing the settings for filtering in related data tables

### Procedure

1. Go to the **Filters** panel and locate the data table header for the data table of interest.
2. Click on the **Filtering in related data tables** icon .
 

If something other than **Ignore filtering** is selected, the icon will be blue. If a relation no longer is valid, the icon will turn red.
3. Select the data table for which you want to change how filtering should affect the current data table, and select one of the options **Include filtered values only**, **Exclude filtered out rows** or **Ignore filtering**.



### Result

The filtering is updated so that when you filter in the first data table, the selected data table will respond in the specified way.



If you also want the filtering to be carried over to the first data table when filtering in the second data table, you must also update the setting from the header for the second data table.

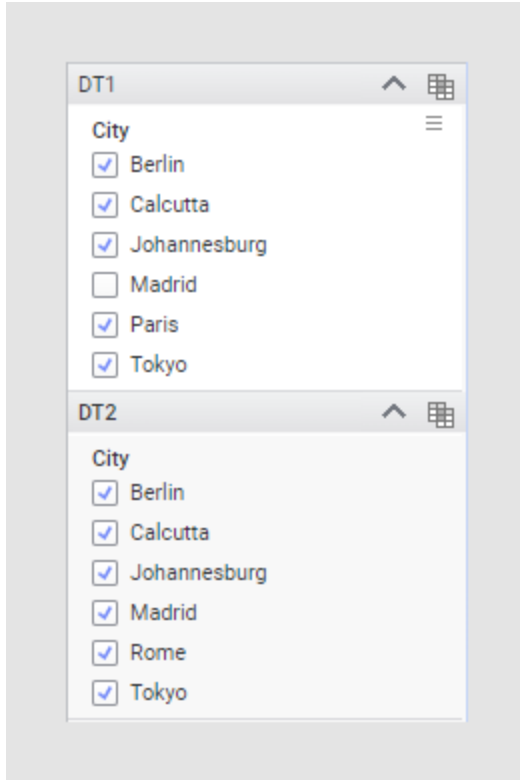


You cannot propagate filtering based on [streaming data](#) to another data table.

### Example: The difference between the settings

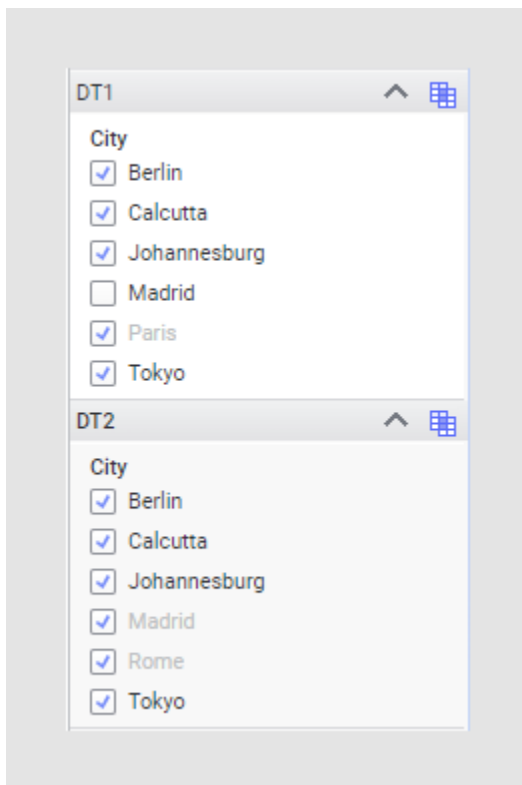
In this example, two data tables, DT1 & DT2, both have a column called City which is used to create the relation. This column shares five values between both tables (Berlin, Calcutta, Johannesburg, Madrid, Tokyo), and each include one additional value that is different in the tables (Paris or Rome).

The default setting is to **Ignore filtering** in the other data table:



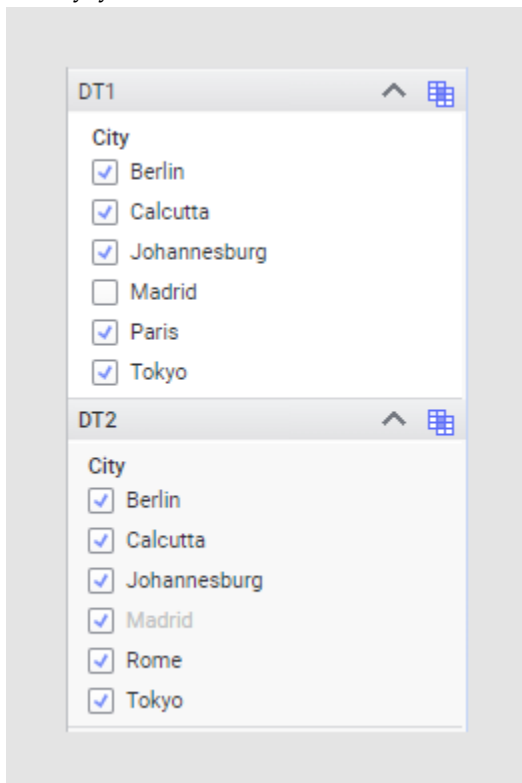
When Madrid is filtered out in DT1, nothing happens to the value in DT2.

If you instead use the **Include filtered values only** option, Madrid is filtered out from DT2 and is greyed out to show that this value no longer is applicable due to the filtering in DT1. Paris and Rome are also greyed out because they are not available in both data tables.



The filter setting is applied in both directions in these examples, but the directions do not affect each other. The result in DT1 is the same, regardless of how DT2 is configured on its end. If you want to include all values from DT1 (that is, to include Paris) but not all values from DT2 (to remove Rome), you would specify the **Include filtered values only** option on DT2, but not on DT1.

Finally, you can use the **Exclude filtered out values** option:



With this option, Madrid is greyed out in DT2, because it has actively been filtered out in DT1, but Paris and Rome remain in DT1 and DT2 respectively, because those values have not been filtered out from any of the data tables.

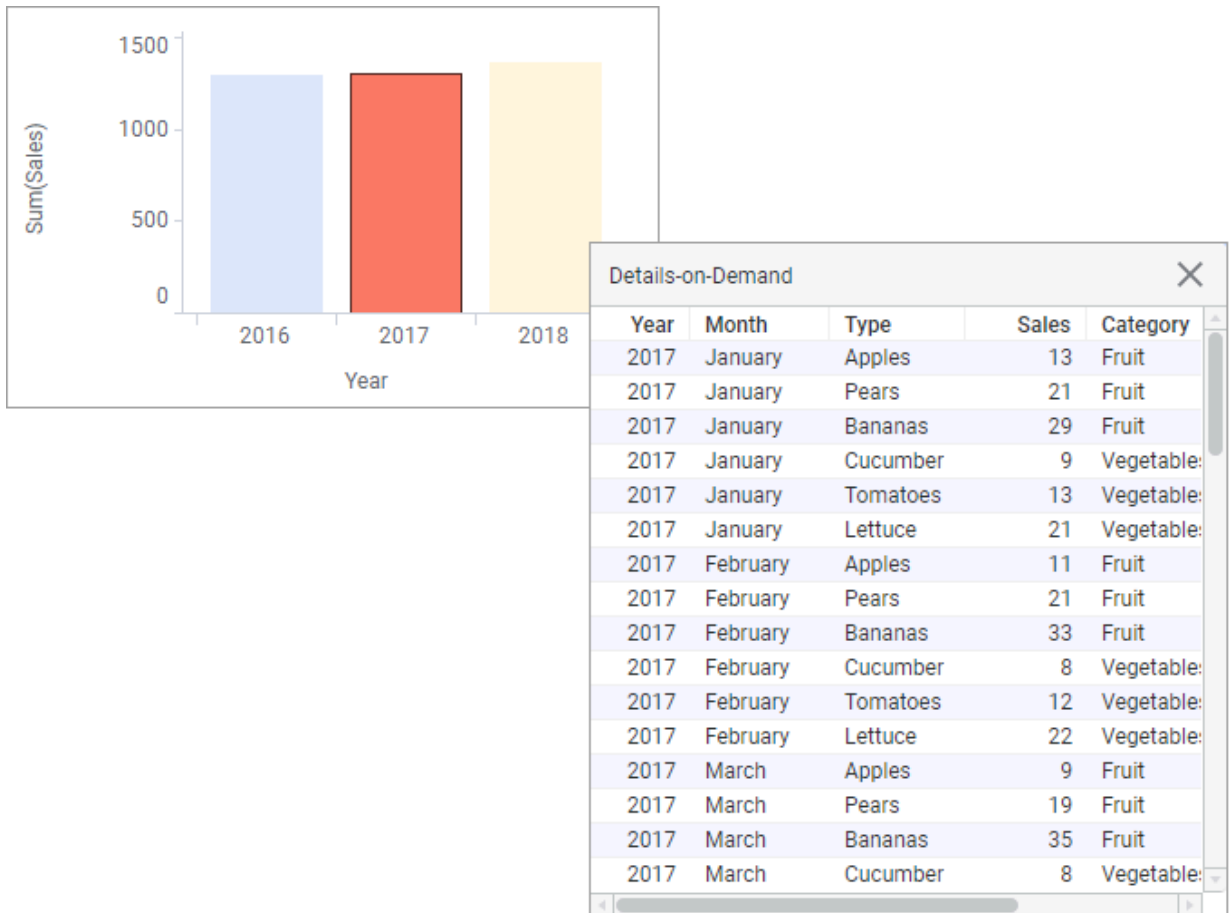
# Analyzing data

You have many opportunities to interact with your data.

## Displaying item details

The actual values of the data rows that are included in visualization items (like bars in a bar chart or sectors in a pie chart) can be viewed in the Details-on-Demand panel.

To select which items the row values should be displayed for, you mark them. If you mark, for example, a bar in a bar chart, all the data rows included in that bar are listed in Details-on-Demand panel as shown below.



### Prerequisites

A visualization is created.

### Procedure

1. If the Details-on-Demand panel is not visible, select **View > Details-on-Demand** on the menu bar. The Details-on-Demand panel is shown on the right-hand side of the page.



- In the visualization, [mark the items](#) which you want to display row details for. The data rows included in the marked items are listed in the panel.



When a visualization combines data from more than one visualization, the Details-on-Demand will show data from the [main data table](#) only. For analyses with multiple data tables, settings for the Details-on-Demand must be defined for each data table.



As long as you work with imported data and you use the Details-on-Demand to view information about more than one row, you can also use the Details-on-Demand as a starting point when [replacing values](#) in a data table.

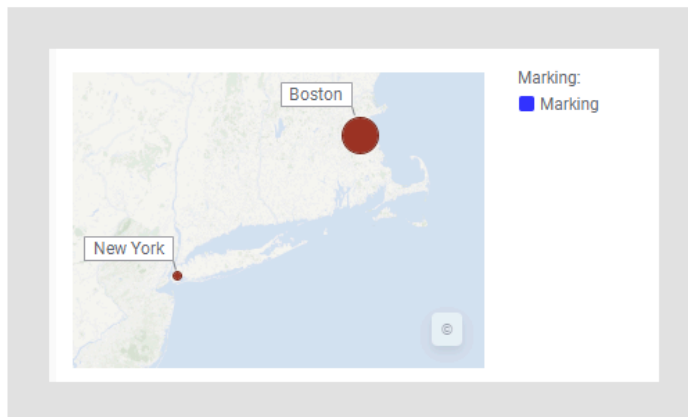
## Drilling down into details

If you want to drill down into specific data in a visualization, you can create a details visualization of a suitable type and configure it to your needs. In the main visualization, you mark the data that you want detailed information about, and in the created details visualization, solely the currently marked data will be shown.



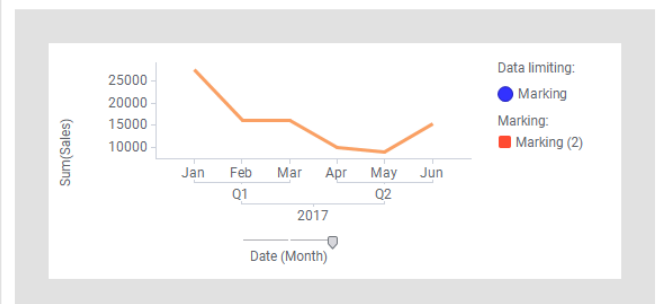
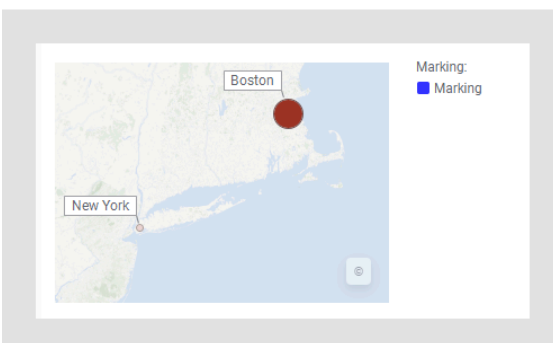
Details visualizations can be used to drill down into your data in multiple steps. The marking you perform in one visualization determines what you will see in the next visualization, and so on.

The map chart below is used to illustrate. The markers represent the total sum of sales at two different stores after the first six months of 2017. However, the data table, which the map chart is based on, contains more information, so it is possible to drill down into details. Assume, for example, you are interested in showing the monthly sales at the Boston store.



Store Location	Date	Sales
Boston	1/2/2017	75
New York	1/2/2017	2053
Boston	1/3/2017	2072
New York	1/4/2017	75
New York	1/5/2017	938
New York	1/6/2017	332
New York	1/8/2017	1309
Boston	1/8/2017	2618
New York	1/11/2017	0
Boston	1/13/2017	364
New York	1/15/2017	4716
Boston	1/16/2017	273
New York	1/17/2017	2508

This can be done by creating a details visualization. The line chart below is such a visualization, but any visualization type can be created. By [marking](#) the Boston marker in the map chart (the main visualization in this example), the monthly sums for the Boston store are shown in the line chart. If you click on another marker in the map chart, like New York, you will instead see the details for New York in the details visualization.





The **Data limiting** setting in the legend shows that the data in the created details visualization is limited by a marking. The line chart above will only show data that is marked in the map chart using this particular [marking](#). Furthermore, a new marking becomes available in the details visualization. This marking will be used to limit the data, if you drill down deeper into the data, using another details visualization. You can read more about data limitations under [Adding data limitations for a visualization](#) on page 555.



It is also possible to view marked data within the main visualization itself. See [Zooming into visualization details](#) on page 601.

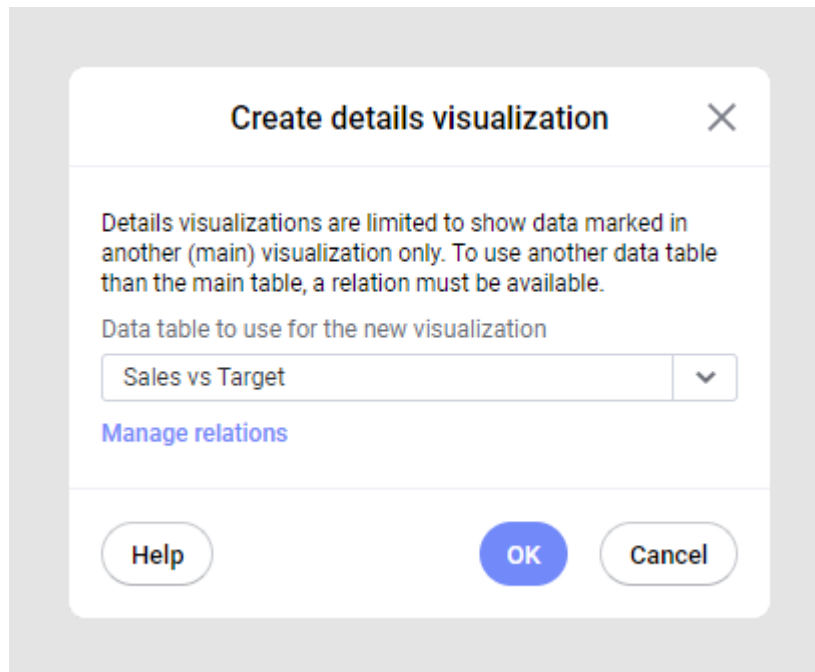
## Prerequisites

The main visualization, that is, the visualization whose details you want to drill down into, has been created.

## Procedure

1. Mark the items, which you want to base the details visualization on.
2. Right-click the visualization, select **Create details visualization** and the visualization type to use as details visualization.

If there is more than one data table in the analysis, and at least two of the data tables have a [relation](#) defined, you can choose to use either the same data table as the one used in the main visualization, or a related data table, for the details visualization. You can also click **Manage relations** to go to the Relations overview in the data canvas.



3. A new visualization of the selected type is presented, based on the marked data only. Adjust the new details visualization according to your needs.



You can [specify what the details visualization should show](#) when no data is marked from the visualization properties.

If you want to drill down deeper and deeper into your data, you can create consecutive details visualizations.

4. Repeat steps 1-3 with the details visualization as starting point.

## Specifying what to show when no data is marked

In a visualization, you can mark data that you are particularly interested in, and then examine it separately in details visualizations. In such details visualizations, you can specify what should be shown when no data is marked in the main visualization.

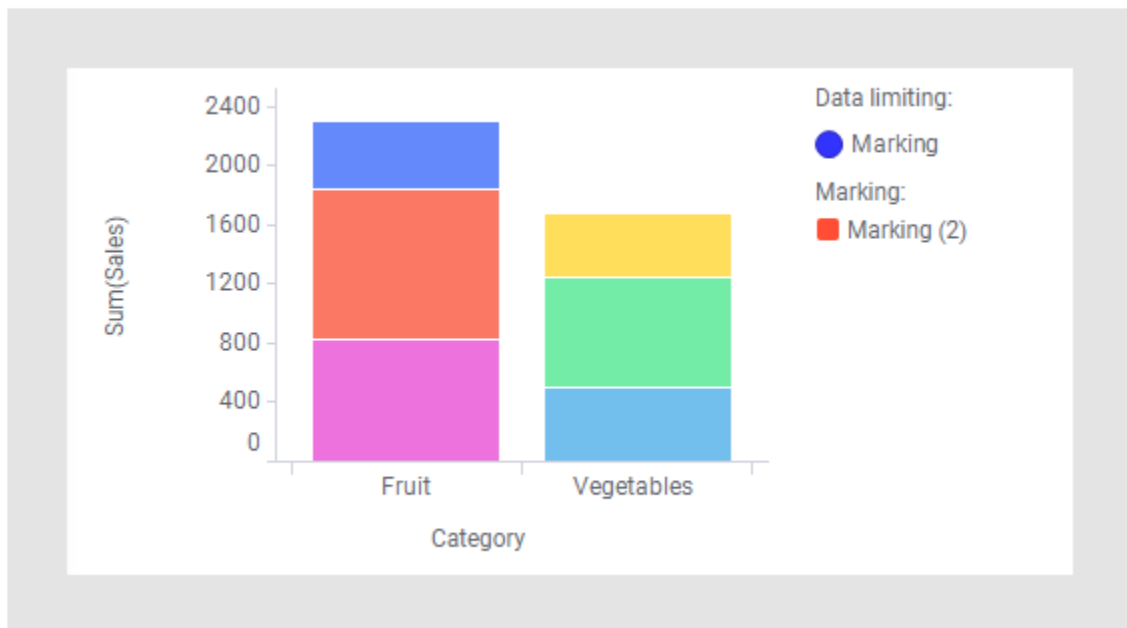
For information about details visualizations, see [Drilling down into details](#).

The following options are available. You can show

- all data
- an empty visualization
- a message on an empty background.

Since different visualization types have different available attributes, it varies between the visualization types exactly what will be shown using the different alternatives. The images below show the three options in a bar chart.

### All data

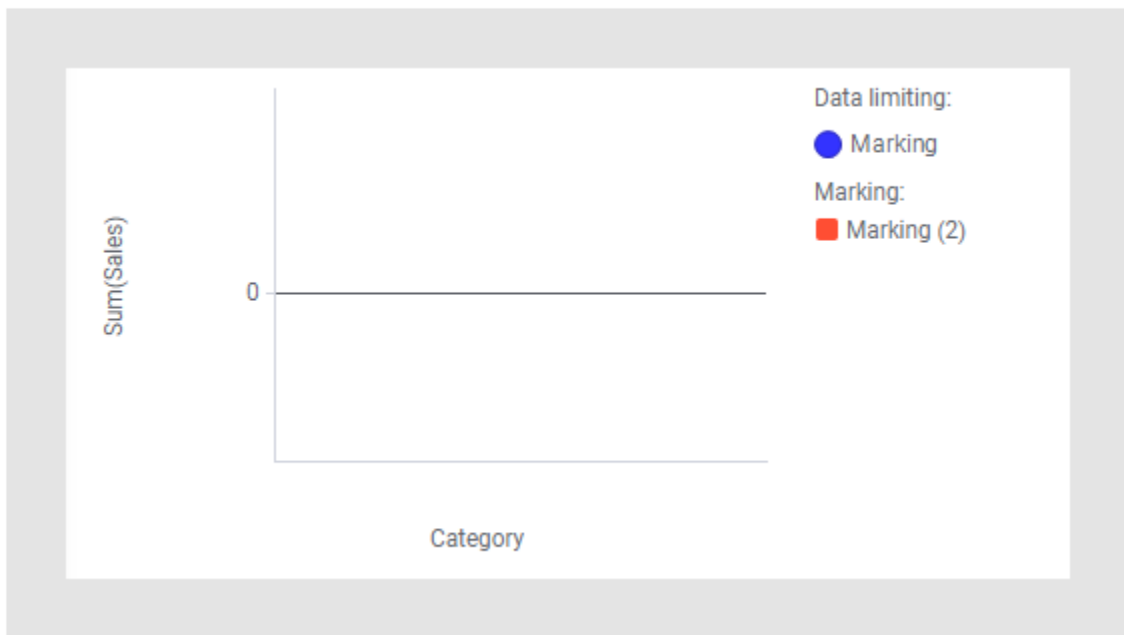


The details visualization will show all the data from the main visualization until you mark something.



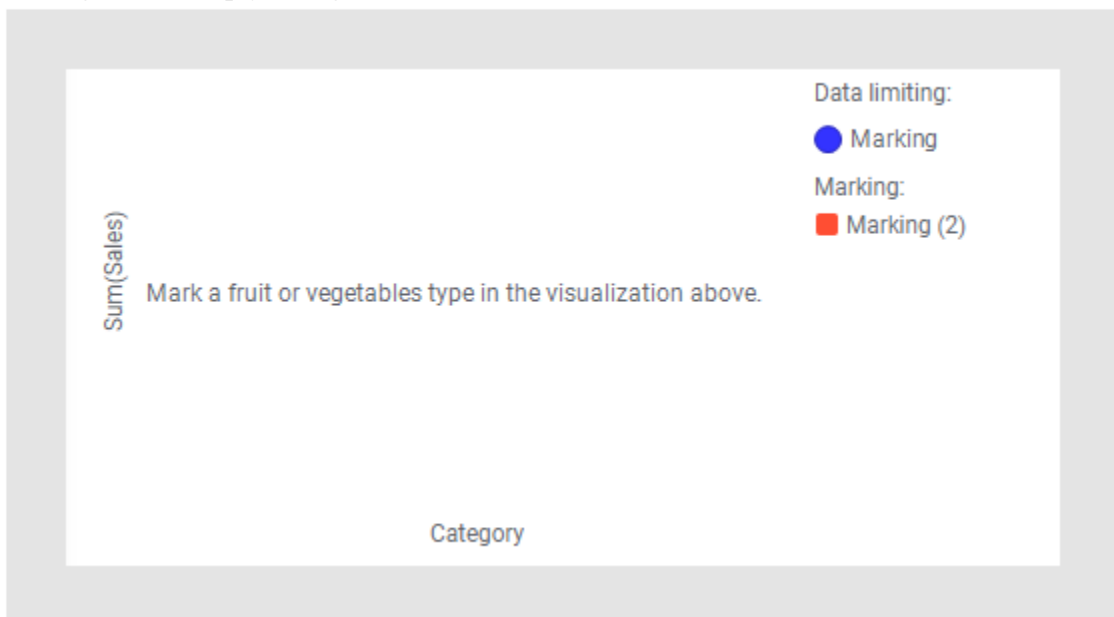
If you use the **All data** option, you can view the marked data in the main visualization itself. For more information, see [Zooming into visualization details](#) on page 601.

### Empty visualization



Unless you mark anything, no data will be shown in the details visualization. Where applicable, axes and scales are shown.

#### Message on an empty background



Unless you mark anything, no data will be shown in the details visualization. You can add a message to let others know in which visualizations to mark items to view details. Where applicable, axes and scales are shown.

#### Prerequisites

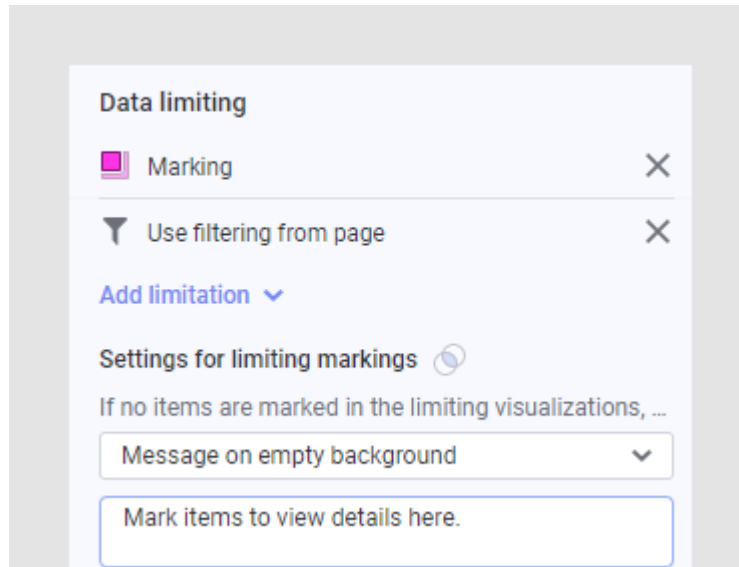
The **Data limiting** setting to limit data using markings in the **Data** section of the visualization properties is used.

#### Procedure

1. In the visualization properties for the visualization of interest, locate the **Data** section.

2. Specify what to show in the drop-down list **If no items are marked in the limiting visualizations, show**.

If you select **Message on empty background**, enter the text to show in the field below the drop-down list.



(The visualization properties look slightly different depending on the client you use and from where you open the properties, but the settings work the same.)

## Zooming into visualization details

You can zoom into visualization details by marking data of interest and view solely the marked data within the visualization.

For a more complete drill-down into the data, see [Drilling down into details](#).

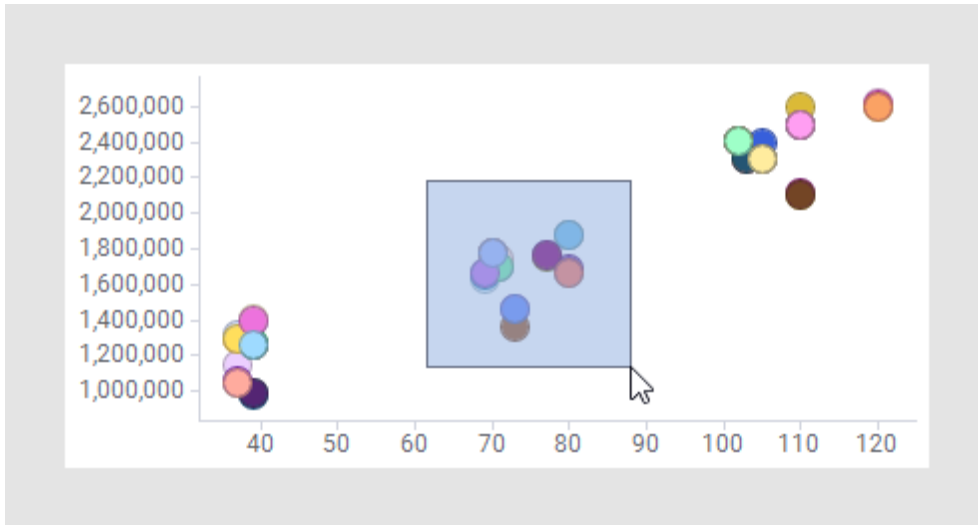
### Procedure

1. In the visualization properties for the visualization of interest, locate the **Data** section.
2. Below **Data limiting**, make sure the visualization's own marking is selected (the marking that is selected in the **Marking** drop-down list in the legend).
3. Under **If no items are marked in the limiting visualizations, show**, select **All data**.

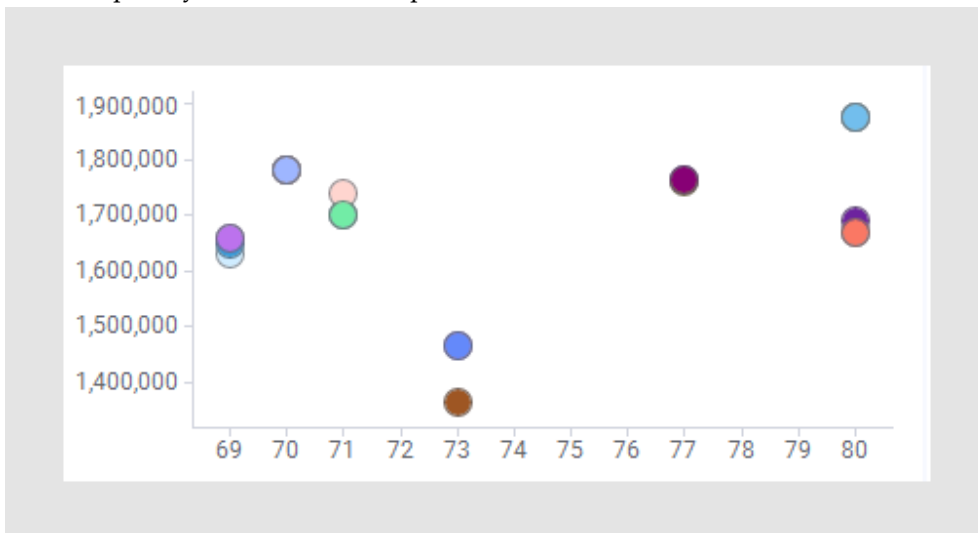
Now you can mark data in the visualization, and the visualization will adapt to show solely the marked data. To return to the original visualization, simply click an empty visualization area.

**Example**

Assume you want to take a closer look at the group of markers in the middle part of the scatter plot below. Follow the steps above, and mark the data of interest.



These steps let you zoom into this particular data as shown below.



# Pages and layout

---

An analysis can be composed in many ways. You can add more pages to the analysis, move visualizations around, change the look and feel using different themes, etc.

## Arranging visualizations

---

You can add several visualizations on a page. Each new visualization will be inserted at the top of the page. Often you will want to adjust the layout of the visualizations, as you might want some to be larger than others, or some visualization to be placed beside another instead of above it.

## Changing the page layout using drag and drop



You can easily resize a visualization, or change the layout of the visualizations by clicking on a visualization title and dragging the visualization to a new position. A shaded box will be shown over the part of the other visualization you are aiming for, and when you release the mouse the layout will be updated as shown below.

### Prerequisites

You must have at least two visualizations on the same page to change the layout.

### Resizing a visualization

To resize a visualization, position the cursor between the visualizations and it will

change its appearance to  or  , then drag to the desired size.

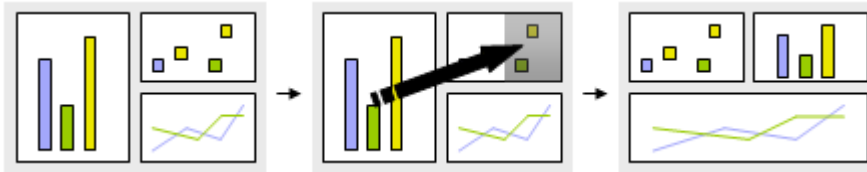


If the visualization becomes too small in either direction, controls such as the legend, zoom sliders, axis selectors, and so on, are hidden and cannot be viewed until the size of the visualization is increased.

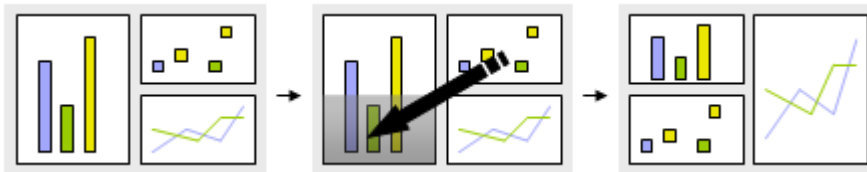
## Moving visualizations

Click the title of a visualization and drag it over another visualization.

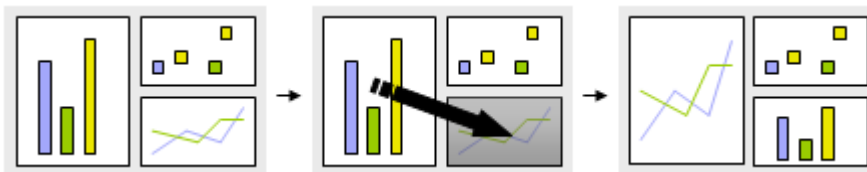
A shaded area will indicate where the visualization can be dropped. All the visualizations will automatically adjust to fill the entire window.



Placing a visualization beside another



Placing a visualization below another



Switching places between two visualizations



If the title bar of a visualization has been hidden, go to visualization properties and select **Show title bar**.



You can drag a visualization to a different page by first hovering with the mouse pointer on another page tab.

## Changing the page layout using arrange visualizations

Working with several visualizations simultaneously can be easier if you arrange them using page layout options.

### Prerequisites


You must have at least two visualizations on the same page to change the layout.

### Procedure

1. On the menu bar, select **Visualizations > Arrange visualizations**.
2. Select an arrangement option from the drop-down menu list.

Option	Description
Evenly	Arranges the visualizations as evenly as possible.
Side-by-side	Arranges the visualizations horizontally.
Stacked	Arranges the visualizations vertically.



Option	Description
<b>Maximize active</b>	Maximizes the active visualization on the page and hides the others. Use the arrows on the title bar of the maximized visualization to view one visualization at a time.  <div>  <div>Maximizing the active visualization might not work with custom visualizations.</div> </div>
<b>Lock visualization area</b>	Fixes a visualization (or several adjacent visualizations) to the left, right, top, or bottom of an analysis page.

## Arranging pages

An analysis can contain one or more different pages. When there is more than one page available, you can navigate between pages in different ways.

### Adding a new page

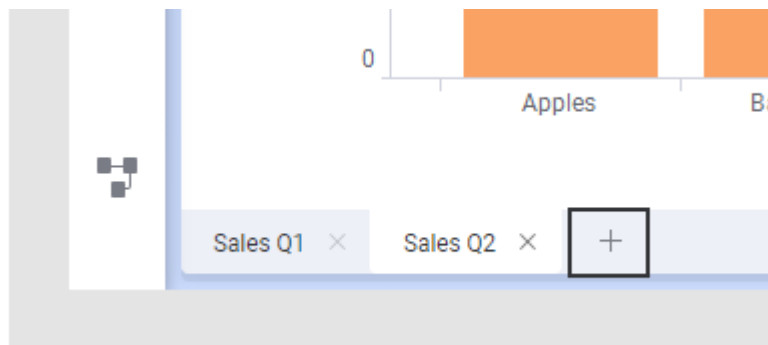
You can work with several pages in your analysis if you want to keep information separated, or if you simply need more space.

#### Prerequisites

The analysis must be in **Editing** mode and you must use the **Titled tabs** [page navigation](#).

#### Procedure

1. Click the plus sign to the right of the page tabs.



An empty page is added to the analysis.

2. If you want, double-click the added page tab, and rename the page.

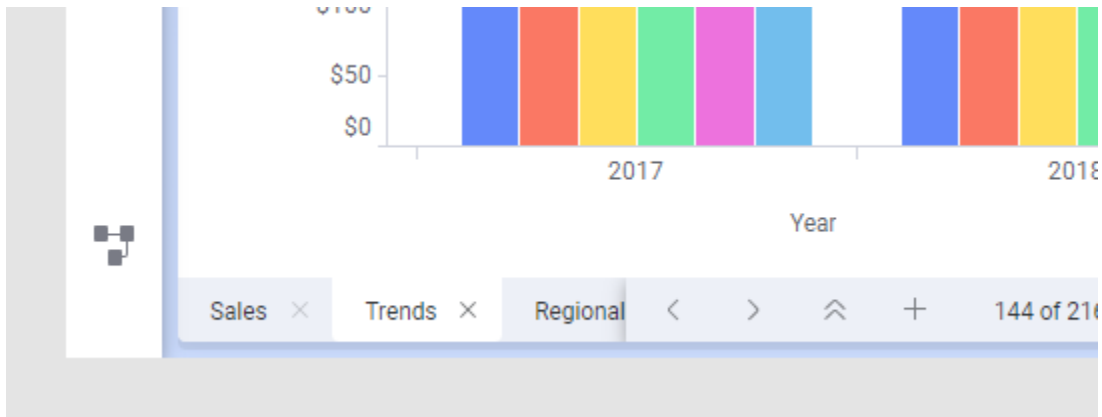
### Changing page navigation method in an analysis

You can add many pages to an analysis, each of them containing one or more visualizations. There are different methods to handle how to navigate between the pages.

The methods are:

- Titled tabs
- Step-by-step
- History arrows
- Off

## Titled tabs



The pages are indicated by titled tabs at the bottom or top of the analysis, and you click these tabs to switch pages. Use this method when you want to see the titles of the pages and be able to easily locate a certain page by its title.

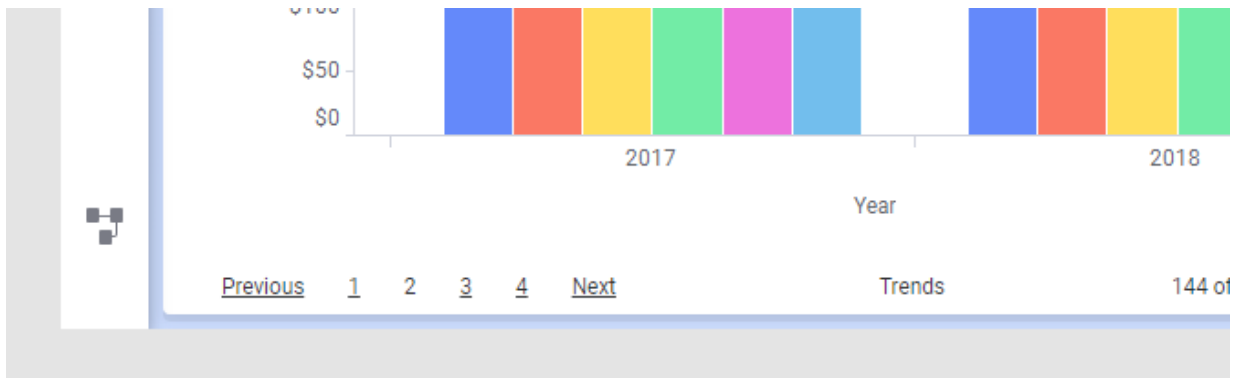
If the analysis contains too many tabs to be shown at the same time, a double arrow is shown next to the plus sign. Click the double arrow to get a drop-down list of all pages in the analysis, from which you can select the wanted page. With the left and right arrows you can scroll horizontally through the pages.

If you want to change the order of the pages, drag the tab to a new position.



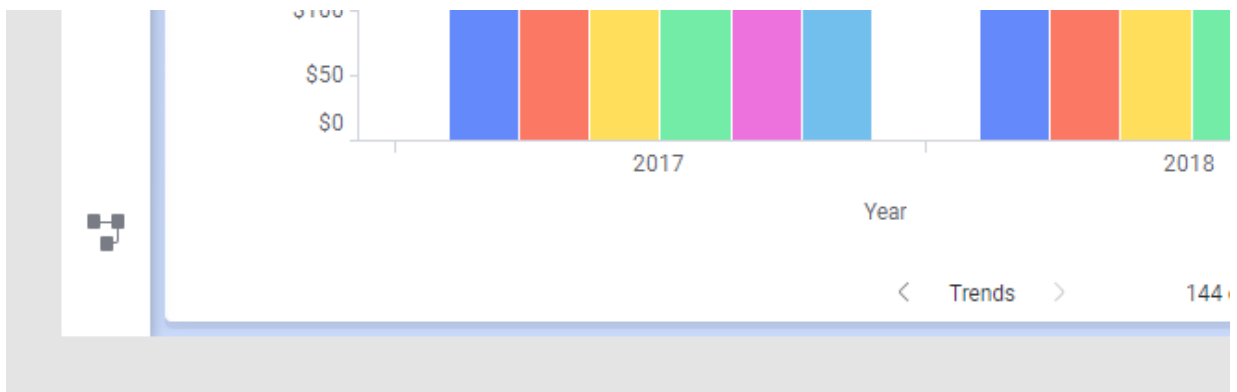
The Titled tabs navigation must be used when you want to add a new page to an analysis.

## Step-by-step



The pages are shown as numbered links, supplemented with Next and Previous links. This method can be useful when the order of the pages is important, and you want to present the analysis in a step-by-step flow.

## History arrows



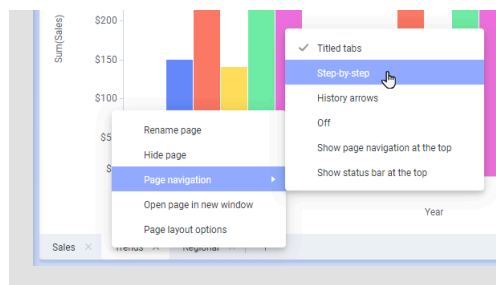
An analysis that is created in the installed client might be configured to guide the user through the pages via various action controls like buttons or links in text areas and certain visualizations. When the navigation is handled this way, the History arrows method might be suitable to use. You see only the name of the current page, and proceed to another page via an action control. However, the page history arrows next to the page title makes it possible to return to a previously visited page.

## Off

You can also switch off all navigation, meaning that no tabs, links, or arrows will be available. In that case, the analysis workflow must be handled using action controls in the same way as described in History arrows (the analysis must have been created in the installed client). You get a cleaner look, which might be beneficial when, for example, there is only one page in the analysis, or you want less cluttered dashboards.

## Procedure

1. Right-click the navigation area at the bottom or top of the analysis.

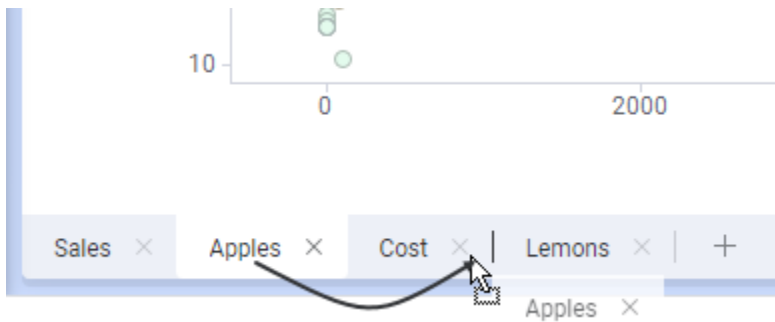


2. Select **Page navigation**, and the navigation method you want to use.

## Changing the order of the pages in the analysis

An analysis can contain more than one page. You can anytime change the order of the pages, as long as the pages are using the Titled tabs page navigation mode.

Drag the page tab sideways to a new position.



## Hiding pages

An analysis might contain pages that, for different reasons, are not intended for Consumers (Consumers use the analysis in Viewing mode). The pages could be in a draft state, or they might contain information that is not relevant for consumers. If you are an author (that is, you have editing rights), you can hide such pages for consumers. This means that consumers will not be aware that these pages exist.

Hidden pages will disappear from the navigation, and they will not be searchable. However, it will still be possible to let consumers access the hidden pages through, for example, bookmarks or action controls.



**CAUTION** Every author will be able to access the hidden pages, and moreover, also consumers, if the analysis is opened in an older Spotfire version. Hence, you should not use this feature on pages that must be hidden due to security reasons.

## Export to PDF

When you do an ad-hoc export to PDF of an analysis that contains hidden pages, the hidden pages will be ignored and treated as if they do not exist. However, it is possible to export hidden pages by including them in a prepared report. Any user, also a Consumer, can export such a report with the hidden pages included.



There might be cases, when it is beneficial to add extra information to analyses to be consumed in PDF format. To achieve this, add the information on certain pages, hide these pages, and then include them in a prepared report.

## Procedure

1. Right-click the page navigation area at the top or bottom of the analysis, and make sure that **Page navigation** is set to **Titled tabs**.
2. Right-click the page tab in question, and select **Hide page**.

In Editing mode, the page tab is dimmed. An author can still access the page by clicking the page tab. When in Viewing mode, the page tab is totally hidden. Since consumers always are in Viewing mode, hidden page tabs will not be visible for them.

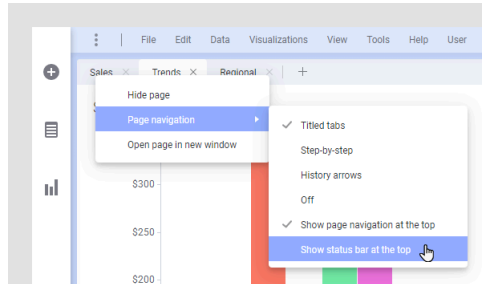
To show the page again, right-click the dimmed tab, and select **Show page**.

## Specifying placement of page navigation and status bar

You can select whether page navigation should be shown above or below the visualizations in the analysis.

### Procedure

1. Right-click the area containing the tabs, links, or history arrows at the bottom or top of the analysis.



2. Select **Page navigation** to open the submenu.
3. Specify your preferred placement using the two menu options **Show page navigation at the top** and **Show status bar at the top** respectively.

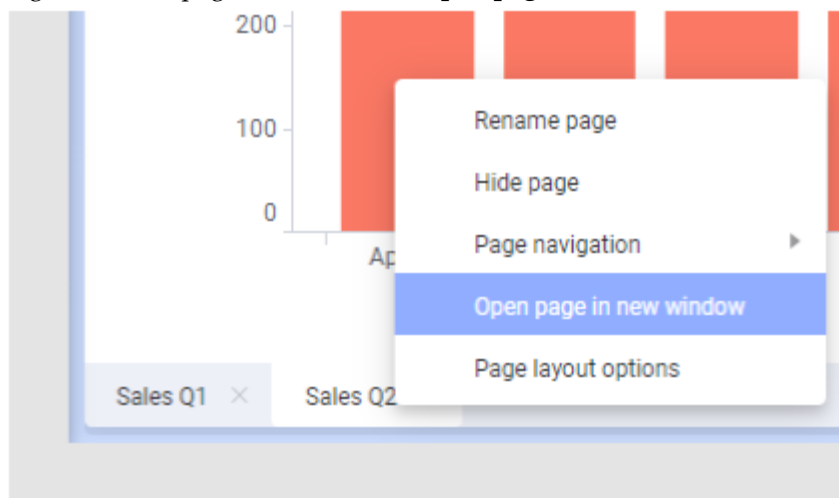
## Viewing analysis pages simultaneously

Pages within an analysis can be viewed at the same time. If you are using the web client you can open a page in a new window in your browser, and then drag the tab, for example, to another screen or next to the already open page in the analysis. With the installed client you can also have multiple windows open with the same analysis. As within any analysis, the pages are connected, so any interactions made on one page affect other pages.

Placing visualizations on different pages, and then viewing them simultaneously, can be beneficial when [exploring data across visualizations](#) or [drilling to details](#).

### Procedure

- Right-click the page tab, and select **Open page in new window**.



### What to do next


Drag the created tab or window to the wanted position on your screen.

## Customizing the toolbar

To the right on the toolbar, you can add shortcuts to your favorite actions and tools.



### Adding shortcuts to the toolbar

1. Right-click the toolbar and select **Customize toolbar**.
2. On the flyout, find the action or tool of interest, then click and drag it to a drop target  in the position you like on the toolbar.

The shortcut is added to the toolbar.



To add many shortcuts to the toolbar, open the flyout, and click each item that you want to add as a shortcut. The items will be added to the toolbar in the order you clicked, and you can rearrange them afterwards. Just make sure the flyout is open until you are done. When the flyout is closed, the shortcuts on the toolbar cannot be moved.

### Removing shortcuts from the toolbar

1. Right-click the toolbar and select **Customize toolbar** to open the flyout.
2. Find the shortcut you want to remove from the toolbar.
3. Drag it downwards and drop it anywhere on the flyout.

The shortcut is removed from the toolbar and returned to its original place on the flyout.

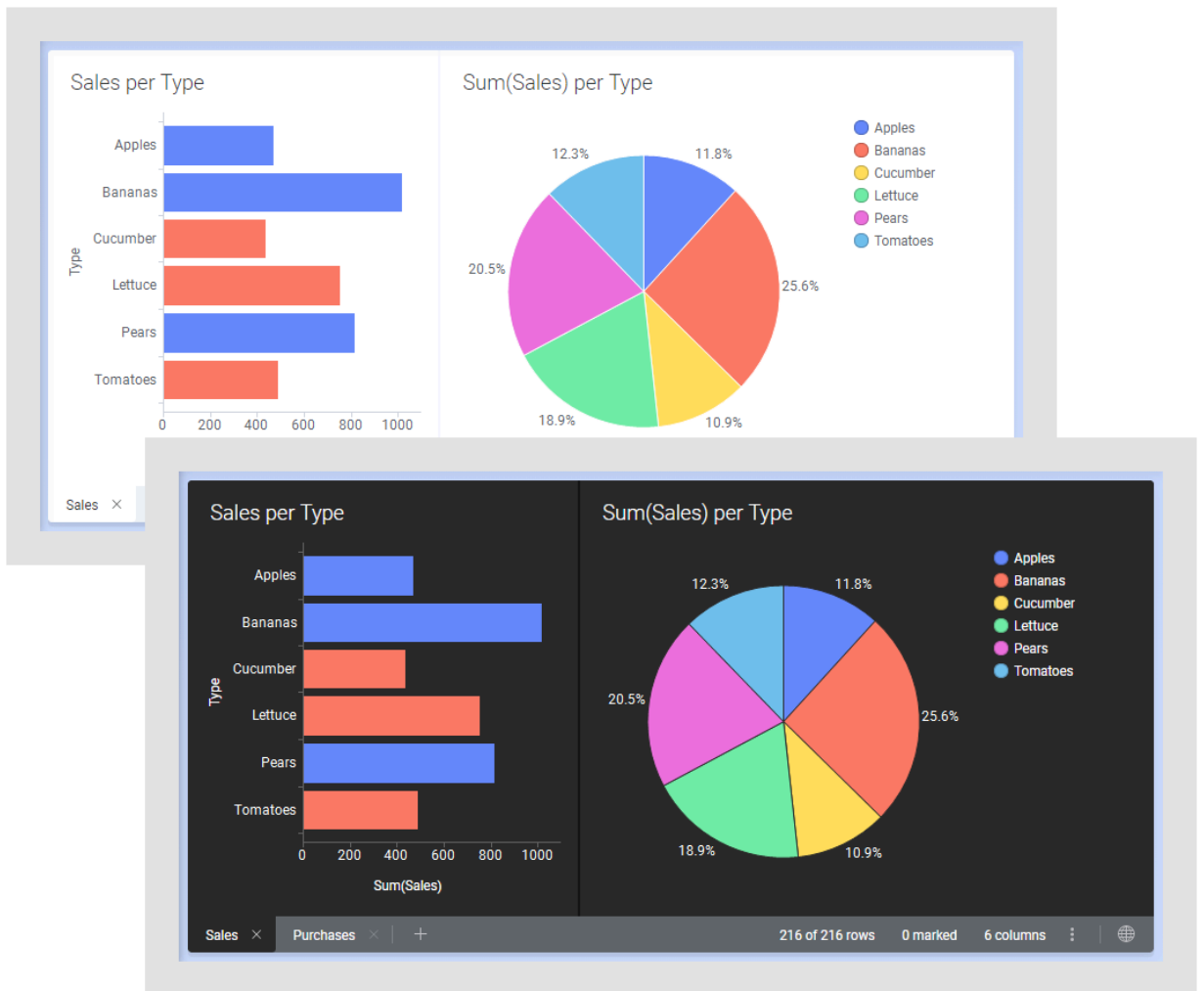


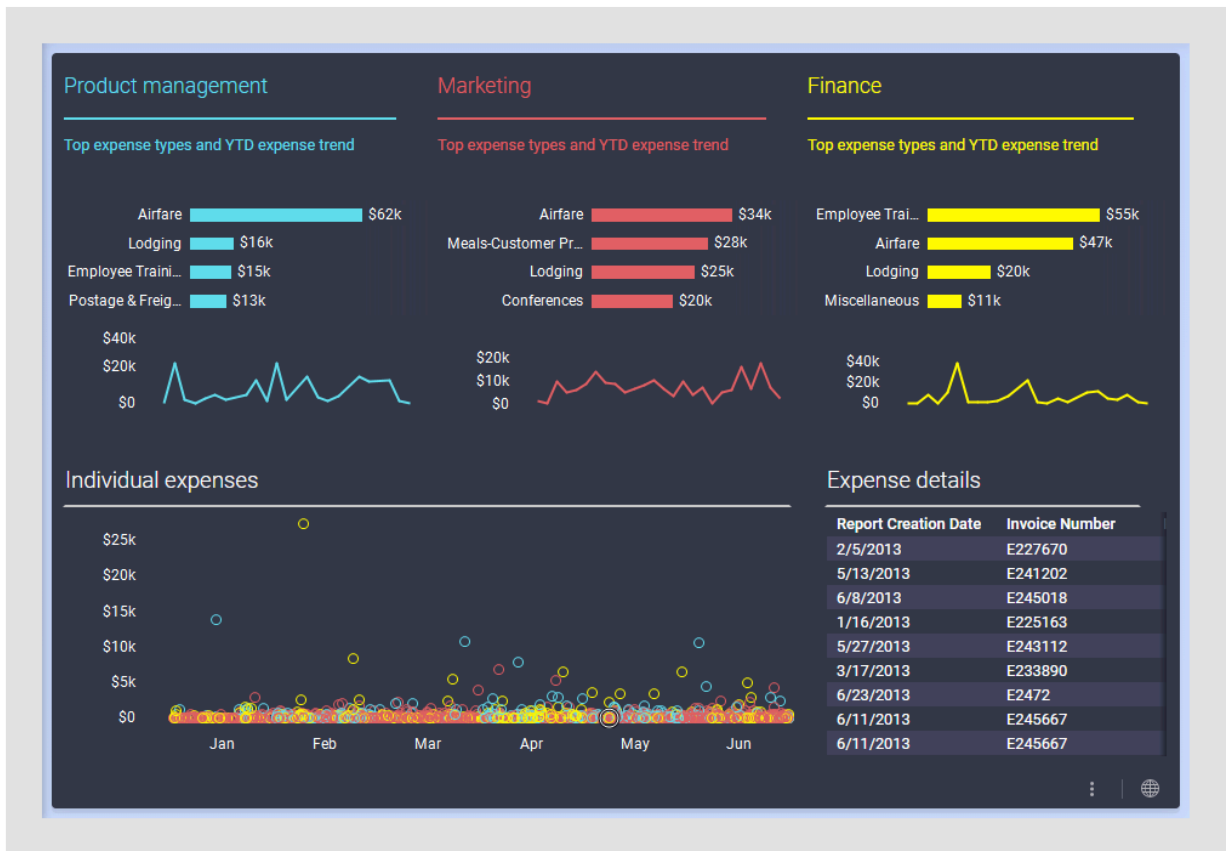
A Spotfire administrator can specify what the toolbar should look like by default, using the `ToolbarShortcuts` preference setting in the Administration Manager (using the installed client).

## Changing the aesthetics

The look and feel of the user interface can be modified. You can use a light or a dark visual theme, or customize your own theme.

Examples of the light and dark visual themes as well as a customized theme are shown below.





## Procedure

1. On the menu bar, select **Visualizations > Canvas styling** to change the look and feel of the user interface.
2. Select a predefined visual theme from the list, or select **Edit custom theme** to define your own visual theme.

## Custom themes

When you customize a visual theme using the Edit custom theme dialog, you can use the light or dark scheme as starting point. Then details such as colors, fonts, borders, and spaces between the objects can be adjusted. This means that you can, for example, create dashboards that follow a corporate graphical profile.

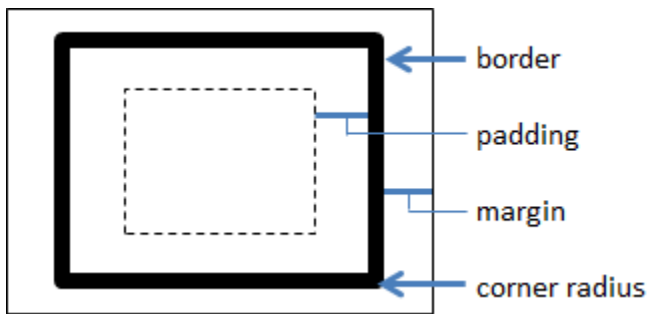
## Custom themes overview

The light and dark visual themes are predefined appearances of the user interface. However, it is possible to customize the visual appearance according to your preferences. Using the light or dark theme as starting point, you can change different visual attributes, for example, colors, fonts, borders, and spaces between objects.

The visual attributes are specified in the **Edit custom theme dialog**.

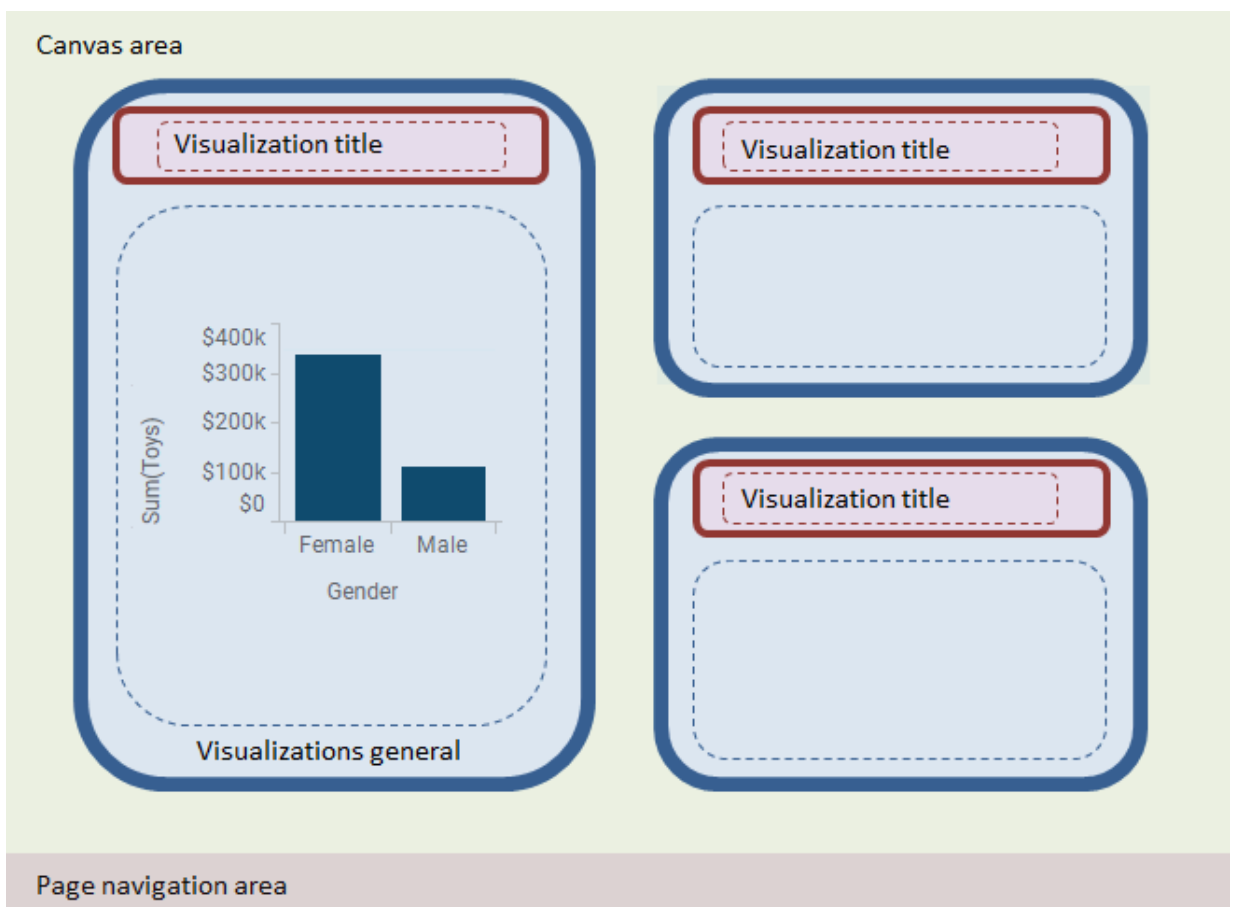
The model below explains some of the objects' attributes that can be adjusted.





The actual content within an object is surrounded by a border. Within this bordered area, you can specify the padding, that is, the width of the space on each side of the content. You can also set border colors, widths and corner radii. Margins to other objects in the user interface can be specified as well.

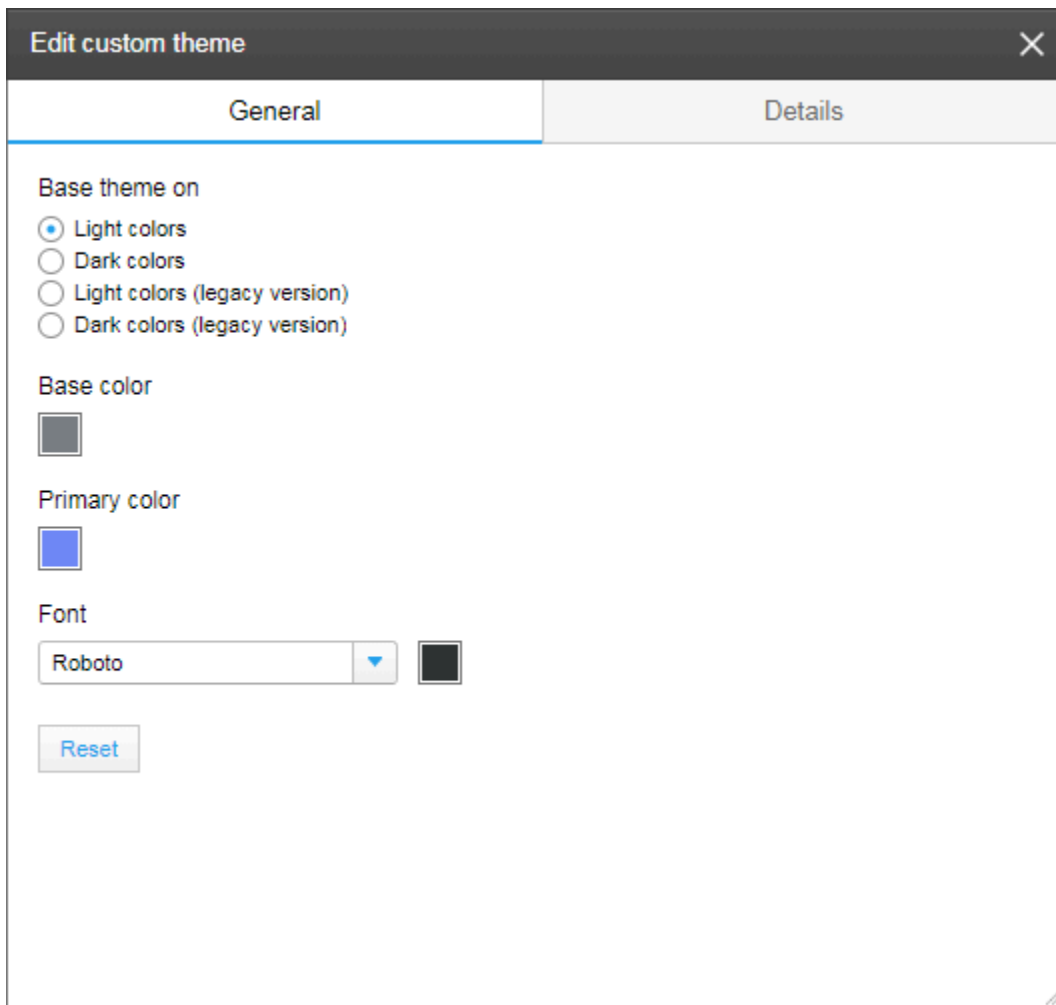
The type of attributes described above can be applied at different places in the user interface like the visualizations and their titles, the page navigation area, annotations, and the canvas area as illustrated below.



## Custom theme settings

In the Edit Custom Theme dialog, you customize the look and feel of the pages. The starting point is one of the predefined visual themes, the light theme or the dark theme.

To reach the dialog, select **Visualizations > Canvas styling > Edit custom theme**.



Because the combinations of visual settings are numerous, experimenting is recommended when customizing a visual theme. You will instantly see the result of a changed setting.

In the **General** tab, you first specify which predefined theme to base your custom theme on, the theme with **Light colors** or the theme with **Dark colors**. (If you want older analyses to keep their schemes, make sure the legacy version is selected.)

You use **Base color** to define the color of label texts, and the general background color in panels and the page navigation area. The **Primary color** defines the color of details like filter sliders or active page indication (the indication is not used by default).

The **Font** selector includes a list of preferred fonts, which can be defined by the Spotfire administrator, to help ensuring that an analysis will look the same for everybody that uses the analysis, including those who are viewing the analysis with web clients.



If you use a font that is not available on the Web Player or Automation Services service host, other users might not get the same experience when looking at the analysis as you do. Fonts might be replaced with other fonts. This is particularly important if the node manager for the service is installed on a [Linux host](#), because the fonts on Linux often differ from Windows fonts. If you want to use a specific font that is currently not included in the **Preferred fonts** section, you might want to discuss with an administrator whether the font can be installed on the node manager host and added as a PreferredFonts preference.

In the different sections of the **Details** tab below, visual attributes can be set for specific parts of the user interface. For more information about the different parts, see [Custom themes overview](#).

Edit custom theme

×

General

Details

Page navigation area

Canvas

Visualizations general

Visualization titles

Visualization scales

Column selectors

Annotations

Font

Roboto

▼

9

▼

■

B /

Background

☐ None

☒ Solid

■

Padding

All sides

12

▲▼

➤

Borders

☐ None

☒ Solid

Border width

All sides

0

▲▼

➤

Border color

All sides

■

➤

Border color for active visualization

■

Corner radius

All corners

0

▲▼

➤

Background color for odd table rows

☐ None

☒ Solid

■

Reset

The image above shows one of the sections, **Visualizations general**. From this section, detailed settings applied to visualizations in general can be made. For example, you can give the visualizations a background color (this affects also the colors of the page navigation area), set the padding, that is, change the distance between the actual content and the visualization border, and customize the appearance of the border, or even remove borders. Note that clicking the arrows offers options to apply a setting on certain sides or corners.

The other sections in the **Details** tab are described below:


- In the **Page navigation area** section, you define the appearance of the page title text and the color to use for indicating the active page's tab.
- In the **Canvas** section, you can change the color of the background on top of which the visualizations are placed. This is also where you specify the distances between the visualizations, and the distances between visualizations and the surrounding panels (padding).
- In the **Visualization titles** section, you define the title bar position and appearance as well as the title text appearance and its position within the bar.
- In the **Visualization scales** section, you define whether or not to show scale lines and tick marks. If shown, their colors can be set.
- In the **Column selectors** section, you control the appearance of the column selector texts.
- In the **Annotations** section, you control the appearance of added annotations.





## Creating a clean visual appearance

A cleaner visual appearance of the analysis below can be achieved using menu options reached from the visualization title bars, by changing settings in the visualization properties, and by using a customized visual theme.

The attributes to modify to get the cleaner look in the bottom image are indicated by numbers, and the actions made are described.



1. Change titles by double-clicking the visualization titles and entering new titles.
2. Remove the legend by moving the cursor over the line chart title, and then click Legend, .

3. Remove selectors in the line chart by moving the cursor over the line chart title bar, and then click  to open the menu. Click **X Axis Selector** and **Y Axis Selector** to remove these selectors. You remove one axis at a time.
4. Remove selectors and scale labels in the bar chart by moving the cursor over the bar chart title bar, and clicking  to open the menu. Then click, one at the time, **Category Axis Selector**, **Value Axis Selector**, and **Scale Labels** to remove them.
5. Keep only the color information in the legend. In visualization properties, locate the **Legend** section and remove all items except **Color by**. Make sure that the **Show title** and **Show axis selector** check boxes are cleared in the settings for the **Color by** legend item.
6. Move color items to the horizontal view. In visualization properties, locate the **Legend** section and remove all items except **Color by**. Click **Show values horizontally**. Alternatively, you can move the cursor over the legend and click  next to the Color by title to move only the color values to the horizontal view. Move the cursor over the horizontal color items, click the option icon , and select **Center** and **Bottom**. Remove the remaining vertical legend as described in step 2.
7. Add labels to the bars in the bar chart. In visualization properties, locate the **Labels** section and set **Show labels** to **All**. Below **Types of labels**, clear the **Bar segments** check box.
8. Get a background without borders and background color. On the menu bar, select **Visualization > Canvas styling > Edit custom theme**. In the **Details** tab, select **Visualizations general**, and set **Borders** to **None**. Select **Canvas** and set **Distance between visualizations** to 0.
9. Remove lines and tick marks on scales. On the menu bar, select **Visualization > Canvas styling > Edit custom theme**. In the **Details** tab, select **Visualizations scales**, and set **Lines** as well as **Tick marks** to **None**.



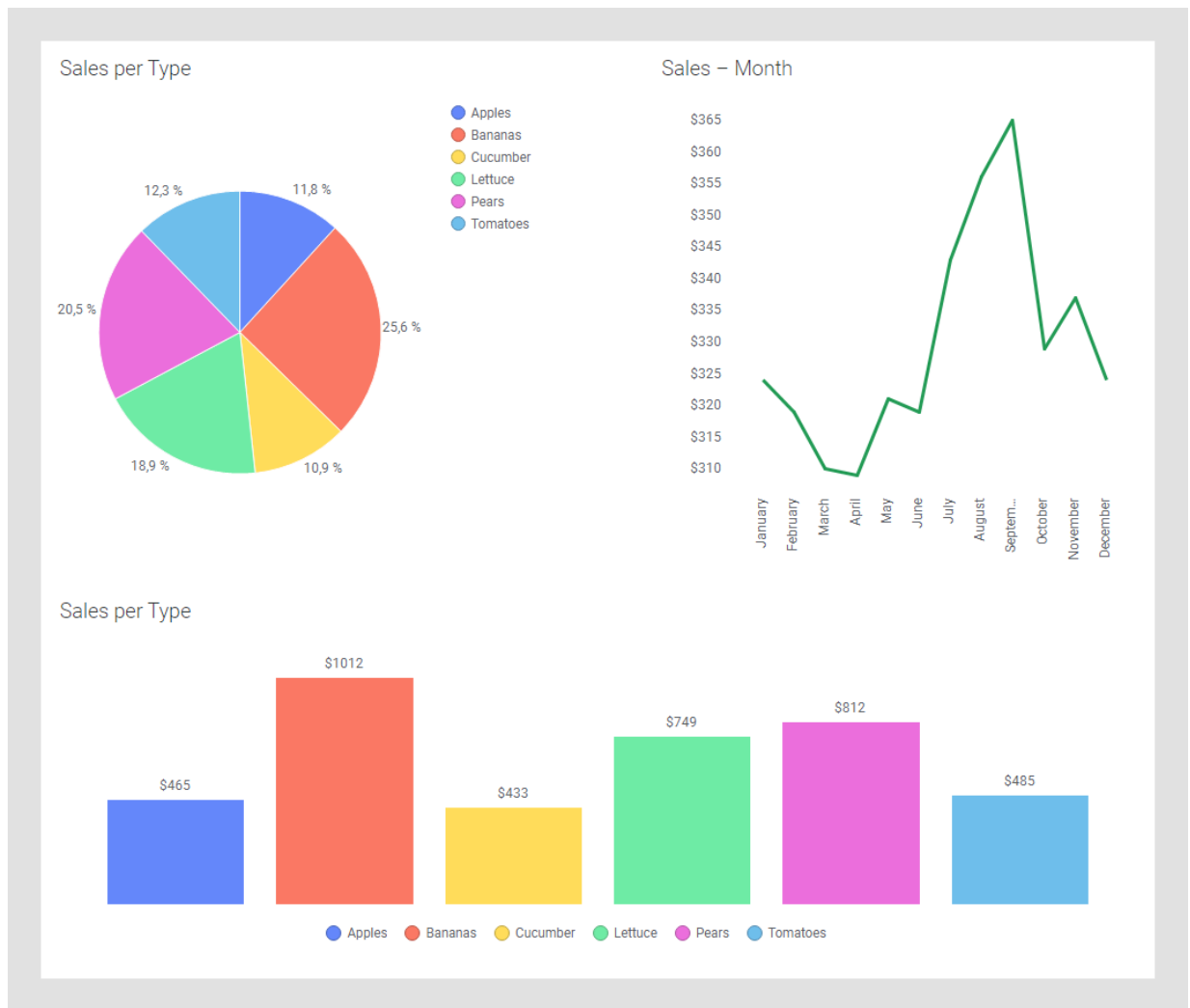
In addition to the numbered actions above, perform an action to separate the visualizations. In the **Details** tab, select **Visualizations general**, and increase the **Padding** number for **All sides**.



To get a more uncluttered look, you can increase padding beneath the visualization titles. In the **Details** tab, select **Visualizations titles**, and click the **Padding** arrow. Then increase the **Bottom** number.

## Result

The dashboard has now a cleaner appearance.



## Adapting layout and panels

You can tailor page layout for various screen sizes, filter on pages and control the visibility of all panels.

### Adapting page layout to different screen sizes

You have different options to determine the size of the area where the visualizations are shown to prepare the analysis for different platforms. For more information, see [Adapting layout to different screen sizes](#) on page 619 and [To specify a particular visualization area size](#) in the *Spotfire Analyst User Guide*.

### Filtering in different pages

You can specify whether or not filtering on one page will affect the visualizations on a different page by using the same or different [filtering schemes](#) on the pages. You can also specify which filters will be visible on each page using the [Organize Filters](#) dialog in the installed client. The filtering scheme of new pages is inherited from the active page, but it can be changed by displaying the [filtering scheme](#) menu in the Filters panel.

## Showing and hiding panels

The visibility of all panels such as the filters panel, and the bookmarks panel is controlled per page. For each page, you can switch on or off the panels of interest by clicking their corresponding toolbar button (and clicking Dock if the panel is in popover mode) or by selecting it from the **View** menu. The panel visibility of new pages is inherited from the active page.

## Adapting layout to different screen sizes

When you create analyses, it is sometimes important to consider what the analysis will look like on screens of different sizes. It is possible to adapt the layout in various ways.

- [Maximizing the canvas](#)

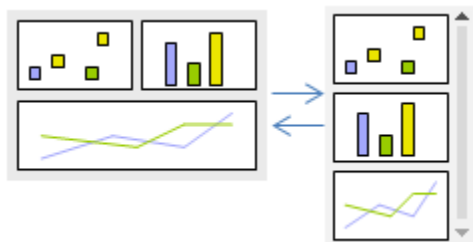
When presenting an analysis to consumers, for example in dashboards on large screens, it might be beneficial to maximize the canvas for display of exclusively visualizations. This can be achieved by hiding the menu and authoring bars, and by removal of the padding around the canvas.

To make even more space available for the actual visualizations, it is possible to enter full screen mode.



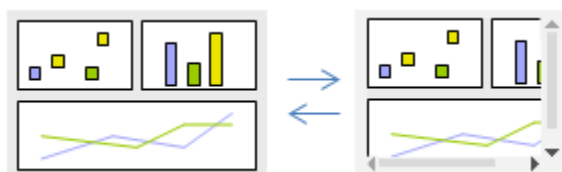
- [Modifying responsive layouts](#) on page 621

The analysis layout is responsive. If the screen width is below a certain breakpoint, there is an automatic switch to a layout, where the visualizations on the page are placed on top of each other, supplemented with a vertical scroll bar. You can change this breakpoint.



- [Specifying minimum acceptable page size](#) on page 622

To ensure the information in the visualizations on an analysis page is readable no matter which screen size is used, you can specify a minimum acceptable width and height of the area used for the visualizations. When the window is resized for any reason, so that the width or height falls below its specified acceptable value, horizontal and vertical scroll bars are added automatically to keep the size of the content.



- [Locking the size of certain visualizations](#) on page 623

Sometimes you might want to avoid scroll bars in text areas, or make sure important content is not hidden. Then you can claim space for these text areas and visualizations by locking their sizes to the edges of the visualization area. Other visualizations will rescale to fit the remaining space.



It is possible to lock areas containing more than one visualization.

- [Authoring analyses for small screens](#) on page 625

When you author an analysis to be consumed on a small screen, for example a cell phone, there are several things you can do to make it a good experience also on a limited screen area.

## Maximizing the canvas

To get maximum display area for the canvas, you can hide the top menu bar, and the authoring bar to the left. The padding around the canvas will also be removed. In addition, it is possible to enter full screen mode to make even more space available for the actual visualizations.



A maximized canvas might be beneficial when displaying dashboards, for example on huge screens, or when you simply want a cleaner look, given that the interaction capabilities provided from the bars are not needed.

### Procedure

1. On the menu bar, select **View > Maximize canvas**.  
The bars are now hidden, and the canvas gets a larger display area.



To show the bars, press the **Esc** key.

2. If you want to display the maximized canvas in full screen mode, press the **F11** key.  
The maximized canvas will cover the entire screen.



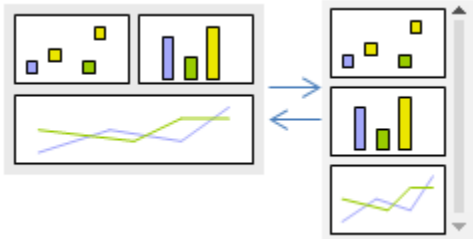
To exit the full screen mode, press the **F11** key once again.



## Modifying responsive layouts

The analysis layout automatically adapts to smaller screens, for example, when the analysis is consumed on cellphones. It is possible to modify this responsiveness for each separate page in the analysis.

The responsiveness means that if the width of the screen is below a certain breakpoint, there will be an automatic switch to a layout, where the visualizations on the page are placed on top of each other, supplemented with a vertical scroll bar.



The order of the visualizations in this single column-based layout is determined by their placement on the analysis page. Visualization areas that have been **locked** are still locked, though, and locking them to the top edge will place them at the top in the column, and locking them to the bottom edge will place them at the bottom.



It is not possible to resize the visualizations using this layout.

The breakpoint, when the switch to or from this column-based layout should take place for an analysis page, can be specified.

### Procedure

1. Make sure the analysis page is active.
2. Right-click its page tab, and select **Page layout options** in the menu.
3. In the Page layout options dialog, make sure the check box **Width less than** beneath **Use mobile layout when** is selected.
4. Specify, in pixels, the breakpoint for the switch. When the width falls below this value, the page layout automatically switches to the cellphone-friendly layout, and if the width exceeds the value, the initial layout is used.



If you want to use the number of pixels of the current width as the breakpoint, click **Set to current width**.

5. Click **Close**.



The **Use mobile layout when** setting overrides any setting made for the **minimum acceptable page size**.

### Details on page layout options

An analysis layout adapts automatically to smaller screens, for example, when using cellphones. It is possible to modify this responsiveness for each separate page in the analysis. You can also specify a minimum acceptable size of an analysis page to keep the content readable. To modify the settings for the layout, right-click a page tab, and select **Page layout options**.

See also [Modifying responsive layouts](#) on page 621.



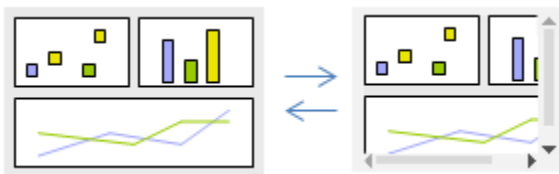
If the canvas size is locked to a specific size, this dialog is not available. For more information, see [Details on Document Properties - General](#) in the *Spotfire Analyst User Guide*.

Option	Description
Use mobile layout when	When you use a mobile layout, the visualizations on the page are placed on top of each other, supplemented with a scroll bar. This layout is used when the screen width falls below a certain breakpoint.
Width less than	Select the check box and specify the breakpoint (in pixels).
Set to current width	Sets the current width of the analysis page as the breakpoint.
Use fixed page size when	When you use a fixed page size, the size of the area containing visualizations is kept also on small screens. Scroll bars are added that makes it possible to view all the content. This layout is used when the screen width or height falls below a certain breakpoint.
Width less than	Select the check box, and specify the minimum width of the visualization area that you accept. If the width falls below this breakpoint, a horizontal scroll bar is added.
Set to current width	Sets Width less than to the current area width.
Height less than	Select the check box, and specify the minimum height of the visualization area that you accept. If the height falls below this breakpoint, a vertical scroll bar is added.
Set to current height	Sets Height less than to the current area height.

## Specifying minimum acceptable page size

To ensure the information in the visualizations on an analysis page is readable no matter which screen size is used, you can specify a minimum acceptable width and height of the area used for the visualizations.

When the window is resized for any reason, so that the width or height falls below its specified acceptable value, horizontal and vertical scroll bars are inserted to keep the size of the content. The breakpoints, when this layout switch should take place for an analysis page, can be specified.



### Procedure

1. Make sure the analysis page is active.
2. Right-click its page tab, and select **Page layout options**.
3. In the Page layout options dialog, beneath **Use fixed page size when**, select the check boxes **Width less than** and **Height less than** respectively to enable specification of the two measures.
4. Specify, in pixels, the width and height breakpoints for the switch of layout.



If you want to use the current width or height of the analysis page as breakpoints, click **Set to current width** and **Set to current height**, respectively.

5. Click **Close**.



The **Use mobile layout when** setting (described on [Modifying responsive layouts](#) on page 621) overrides the settings made for the minimum acceptable page size.

## Locking the size of certain visualizations

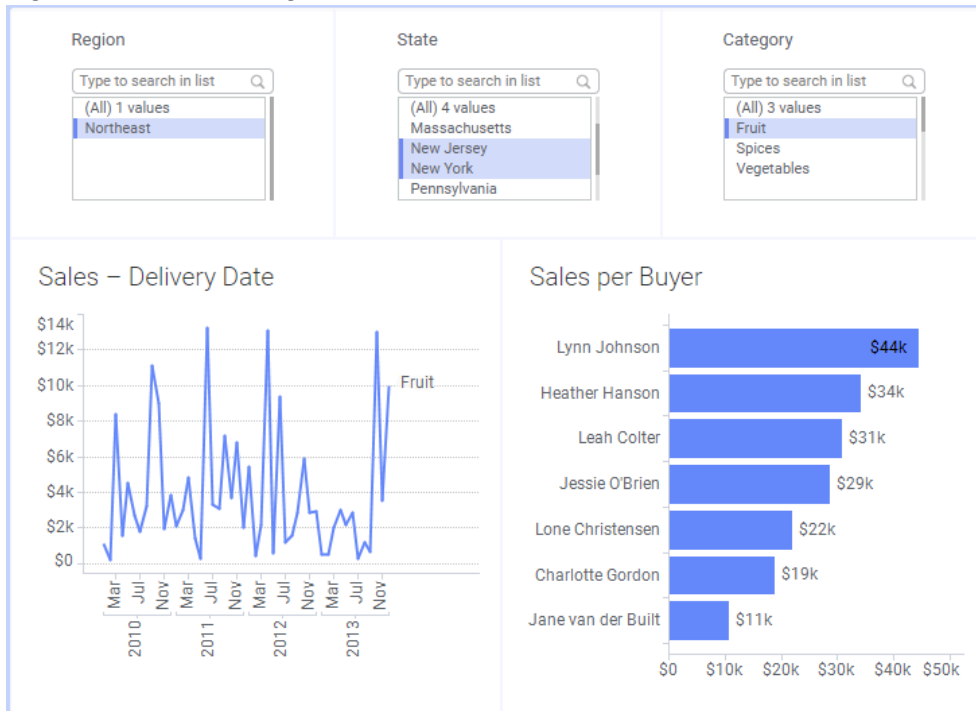
It is sometimes important to consider what the analysis will look like on screens of different sizes. For example, you might want to avoid scroll bars in text areas, or make sure important content is not hidden. One way to control the layout, is to lock the size of certain visualizations.

### Procedure

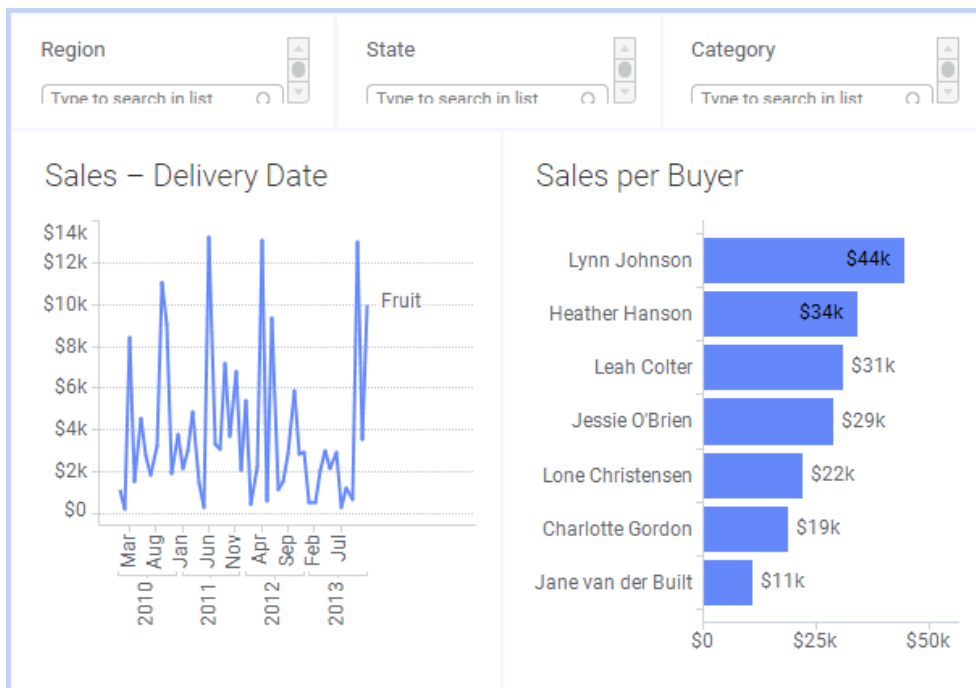
1. Select **Visualizations > Arrange visualizations > Lock visualization area** on the menu bar.  
The visualizations are dimmed out, and padlocks are shown between the visualizations where it is possible to lock one or more visualizations to the edge of the analysis page.
2. Click a padlock next to the visualizations you want to lock.  
A pop-up menu is displayed, showing in which directions it is possible to lock the visualizations (top, bottom, left, or right of page).
3. Select an option from the menu.  
The area that will be locked is highlighted.
4. Click anywhere to return to the visualizations.  
The visualizations within the selected area will now keep the height or width, depending on which direction the visualization area was locked.

## Example

The image below shows an analysis with two visualizations displaying sales data in the lower part of the analysis page. Three text areas are located in the upper part of the page. Each text area contains a filter, making it possible to choose which regions, states, and categories to view in the visualizations below.

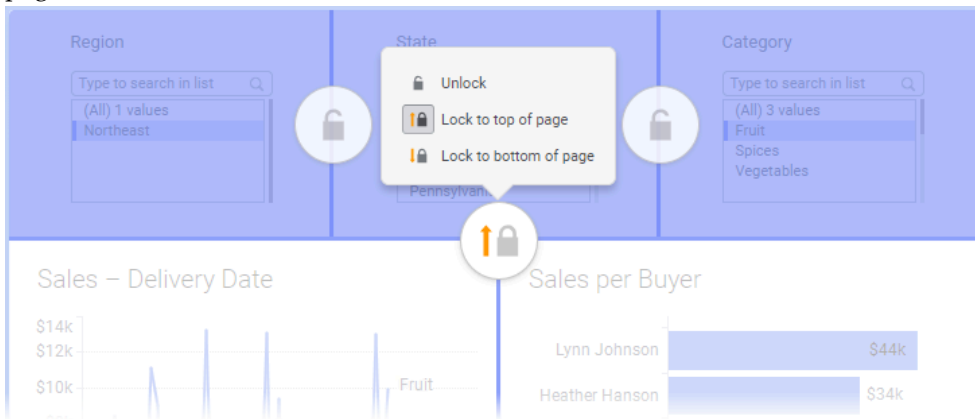


The image below shows the same analysis opened on a smaller screen, without any locked visualization areas:

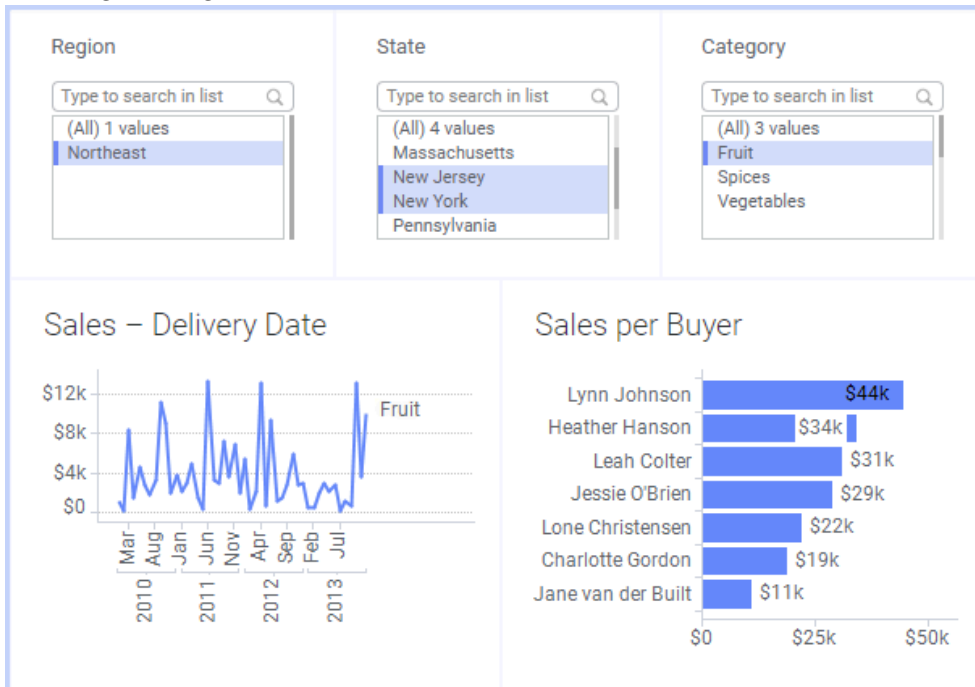


All the visualization areas have been resized to fit the smaller screen, and the text areas containing filters therefore have scroll bars, which makes the analysis look ugly and difficult to use. It would be better if the text areas kept the height they

had in the original analysis. To avoid scroll bars in the text areas, you can lock the upper visualization area containing the filters, to the upper edge of the analysis page.



The upper visualization area has now been locked, and the text areas will keep their original heights also on smaller screens:



## Authoring analyses for small screens

When you author an analysis to be consumed on a small screen, for example a cell phone, there are several things you can do to make it a good experience also on a limited screen area. Some of these suggestions can only be done in the installed client.

- Adjust your screen size to approximate the size and shape of the small-screen device. Take into account that the web browser control display also will take up space.
- Hide panels and controls that are not necessary for the understanding of the analysis.

- Fit texts to the smaller area, for example:
  - Adapt fonts to suit the smaller screen. Use **Tools > Options > Fonts** in the installed client for a global resize.
  - Rename columns to shorter titles.
  - Use short number formats.
  - Show as few decimals as possible.
  - Consider whether horizontal or vertical orientation of labels is most suitable.
- If a visualization title is redundant, remove it.
- Think of making interaction controls easy to use by adult-sized fingers. However, design the analysis to require as little selection, tapping, or dragging as possible.
- To show summaries or trends, consider using [KPI charts](#) (authored using the installed client).
- When possible, [use a text area](#) (authored using the installed client), preferably placed at the top of the page, for interaction controls to provide a more user-friendly experience, and for guiding the user through the analysis. You might:
  - Add buttons for going back and forth between the analysis pages (the default buttons in the browser might be difficult to touch).
  - Add action controls to handle filtering.
  - Consider using dynamic controls such as sparklines, calculated values, icons, and bullet graphs to provide direct summaries of data, but also to provide navigation possibilities.
  - Specify suitable font style and size, and make sure the padding around controls is sufficient.
- Adapt the default layout [responsiveness](#) to the intended screen, and consider [locking visualizations](#) to, for example, avoid scroll bars in text areas.
- Last, but not the least, test the analysis on the small screen.

## Panels and popovers

---

In Spotfire, some functionality can be reached through panels that can be shown either at all times or on demand only. The state of a panel is in many cases remembered per page, so you can use one state on one page and another on the next page.

By default, the [Filters](#) panel and [Details-on-Demand](#) are shown as docked panels to the right and the [Bookmarks](#) panel, the [Tags](#) panel, the [Lists](#) panel and the [Web page](#) panel will be shown to the left in the main window, if displayed. However, some of these panels can be shown in two different states: as docked panels and as popovers.

The [Markings](#) panel, the [Filtering schemes](#) panel and the [Document properties](#) panels cannot be docked.



Some of the panels are only available in authoring mode and some panels are only available in the installed client.

No matter what state is used to display the panel, the content will always be remembered.

### Docked panels

Docked panels might be preferred when you interact with the content of the panel often, or, when you need to see the content at all times, for example, if you need to view the current state of the filters in the filters panel. Docked panels can be moved to the left, right or bottom part of the Spotfire window using drag-and-drop.

### To move docked panels inside the Spotfire main window:

1. Click the title bar of the panel you want to move.
2. Drag the panel to its new position.


If no other panels are shown where you want to place your panel, simply drag it to the edge of the main window and a shaded area will show you where the panel will be placed. If there is no shaded area you will not be able to drop the panel in that position.

If one or more panels are already shown where you want to place your panel, the shaded area will help you decide the order of the panels.

### Popovers


Popovers are useful when the screen estate is limited and you only need a short interaction with the controls within the panel. For example, if you want to apply a bookmark and then continue with your analysis.

### To undock a panel into a popover:

Click the **Undock** icon, , in the top right corner of the panel. The panel will then change into a popover.

You can click the title of the popover and drag it to another position, temporarily. The popover will remain open on this position as long as you work within it. However, the next time the popover is opened it will be relocated to its default position.

### To turn a popover into a docked panel:

Click the **Dock** icon, , in the top right corner of the popover. You can also double-click on the header of the popover to dock it.

## Web page panel

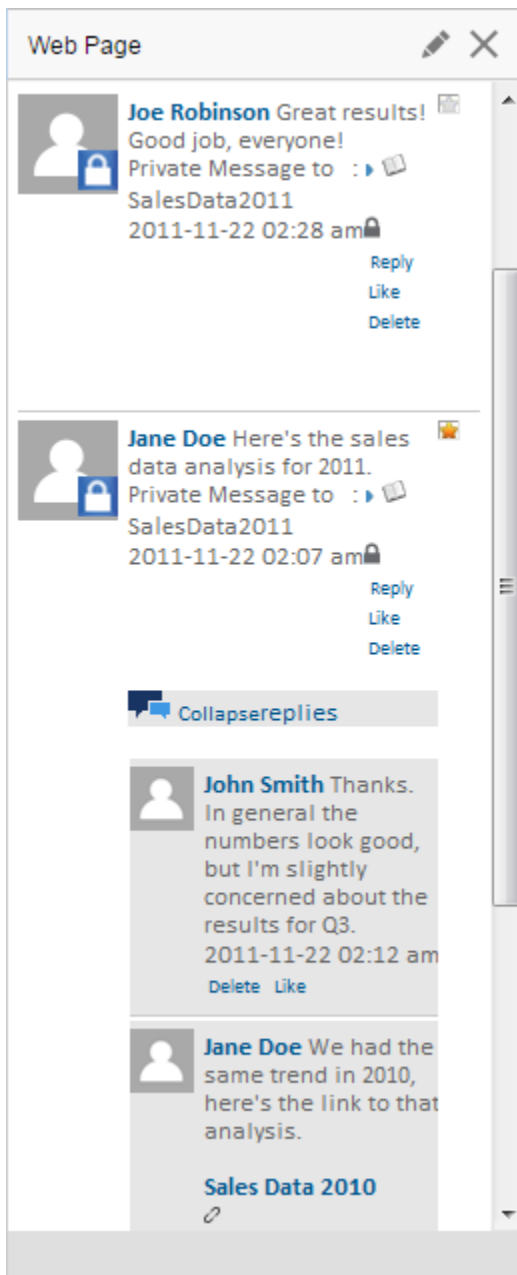
The Web page panel is a tool that allows you to view web pages in Spotfire. This is useful, for instance, if you use some kind of web-based tool to communicate at your workplace, enabling users to work together by participating in communication threads about different subjects.



If you use TIBCO Cloud™ Spotfire®, the Web page panel is not available.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.



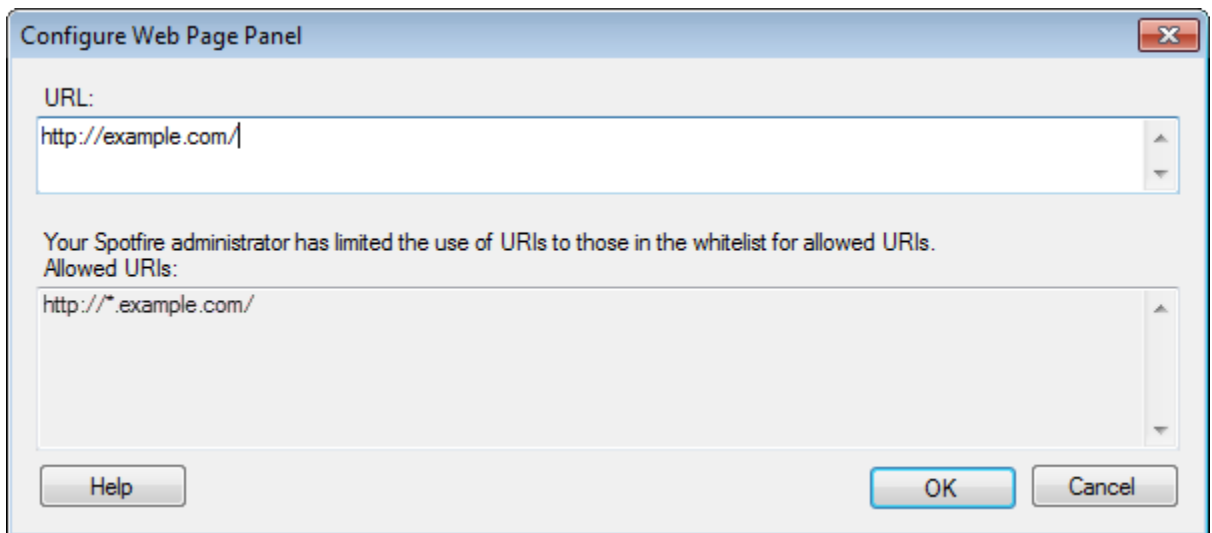
To configure the Web page panel, you enter a URL to a web page that Spotfire will show in the panel (in the installed client).

In the installed client, the Web page panel will run an instance of the Chromium Embedded Framework installed on the computer to parse content. In web clients, the panel will open in an iframe and use the same rendering function available to the rest of the web client. If the Web page panel is used in both the installed client and the web client, this could potentially lead to differences in html rendering between different web browser versions.



The Web page panel is configured per page in the analysis, so you can link the Web page panel on different pages to different URLs. This way, it can be possible to follow several different subjects within a single analysis.





Type or paste the URL to the web page of interest in the text box.



Your Spotfire Administrator might have specified that only certain URIs should be allowed to link to. If that is the case, then a list of the allowed base URI addresses will be shown in the dialog. An administrator can add more addresses to the `Whitelist for Allowed URIs` preference found under `ApplicationPreferences` in the Administration manager. If an allowed URI includes a wildcard character (for example, `http://*.example.com/`), then you can use any path within that specified domain.

## Using the Web page panel

The Web page panel is a tool that allows you to view web pages in Spotfire. If the panel is enabled in your organization, you can show it by selecting **View > Web page**.


### Prerequisites

The Web page panel must be configured in the installed client.



If you use TIBCO Cloud™ Spotfire®, the Web page panel is not available.

### Procedure

1. On the menu bar, select **View > Web page** to show the panel.
2. To configure the panel, click the Configure button, , at the top of the panel.
3. Paste the URL of interest in the dialog.



The Web page panel is configured per page in the analysis, so you can link the Web page panel on different pages to different URLs.

4. Click **OK**.

# Extending Spotfire

---

Spotfire is highly extendable and customizable. The Spotfire environment contains a variety of options to extend the core functionality with custom components, you can customize and automate your Spotfire analyses or even embed Spotfire visualizations into other web applications.

Learn more on the Spotfire community, under [Extending Spotfire](#).

You can extend Spotfire yourself using Spotfire publicly published APIs, download extensions from the [component exchange](#) or one of our partners marketplaces, or engage a Spotfire partner to create it for you.

There are some ways to add functionality to an analysis which are easier than creating actual custom extensions.

## Spotfire mods

The Spotfire environment allows you to enhance the base functionality with your own custom visualizations, or scripts that can perform different actions. The mods framework allows you to develop and debug the mods within the context of a Spotfire analysis, and includes the important functionality to sign and trust the custom code out of the box.

See [Developing mods](#) on page 630 for more information.

## Data functions

A business author might have access to data functions that have been created by a script author using Spotfire Analyst and saved in the Spotfire library. These data functions can be used to add more advanced calculations and use scripts based on different programming languages in the analysis.

Data functions can be used for various purposes like opening data, performing a transformation, or performing calculations like adding clustering or other statistical methods. There are a number of data functions available on the [Spotfire Exchange](#) page on the Spotfire Community.

See [Data functions and the f\(x\) flyout](#) on page 662 for more information.

## Actions and scripts

Actions is a Spotfire concept that offers a wide range of possibilities to perform tasks from within an analysis, including the possibility to run scripts, and they can also include tasks that affect external systems. The actions can be added to text areas and visualizations, and they are initiated using various controls or triggers.

See [Adding an action trigger to a visualization](#) on page 649, [Adding actions to a text area](#) on page 636, [Adding actions performed when clicking on visualization items](#) on page 646 and [Adding a new external action to an analysis](#) on page 650 for more information.

# Developing mods

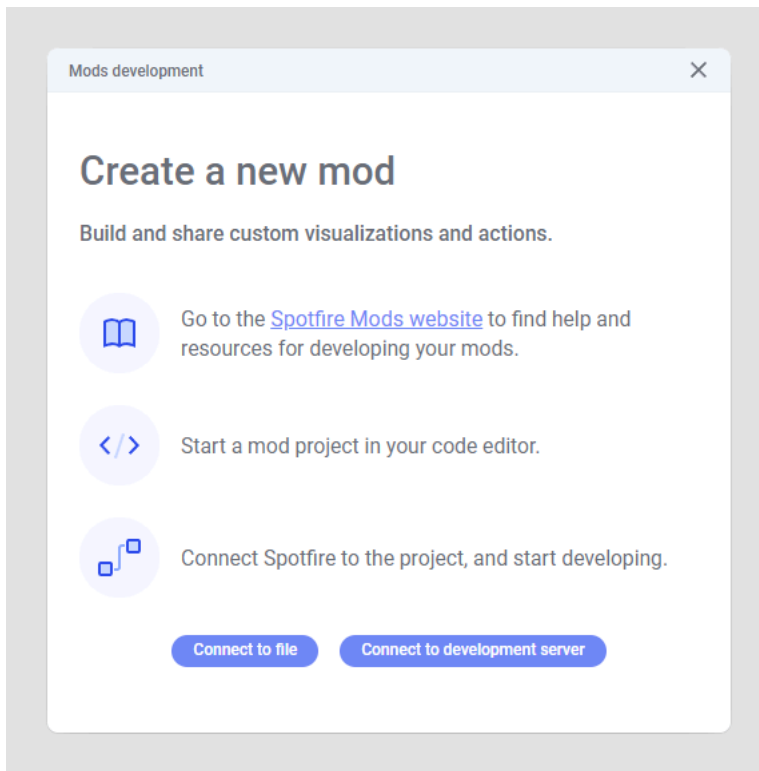
---

The Spotfire environment allows you to enhance the base functionality with your own custom visualizations, or scripts that can perform different actions. To provide a secure development and usage environment, the mods framework has been created.

When mods of different types are developed, they are signed by the author, and the company or the end user, can determine whether they want to trust the custom code, based on their trust in the signer.

On [the Spotfire® Mods website](#) on GitHub, you find complete information about mod development such as prerequisites, a getting started guide, and technical details.

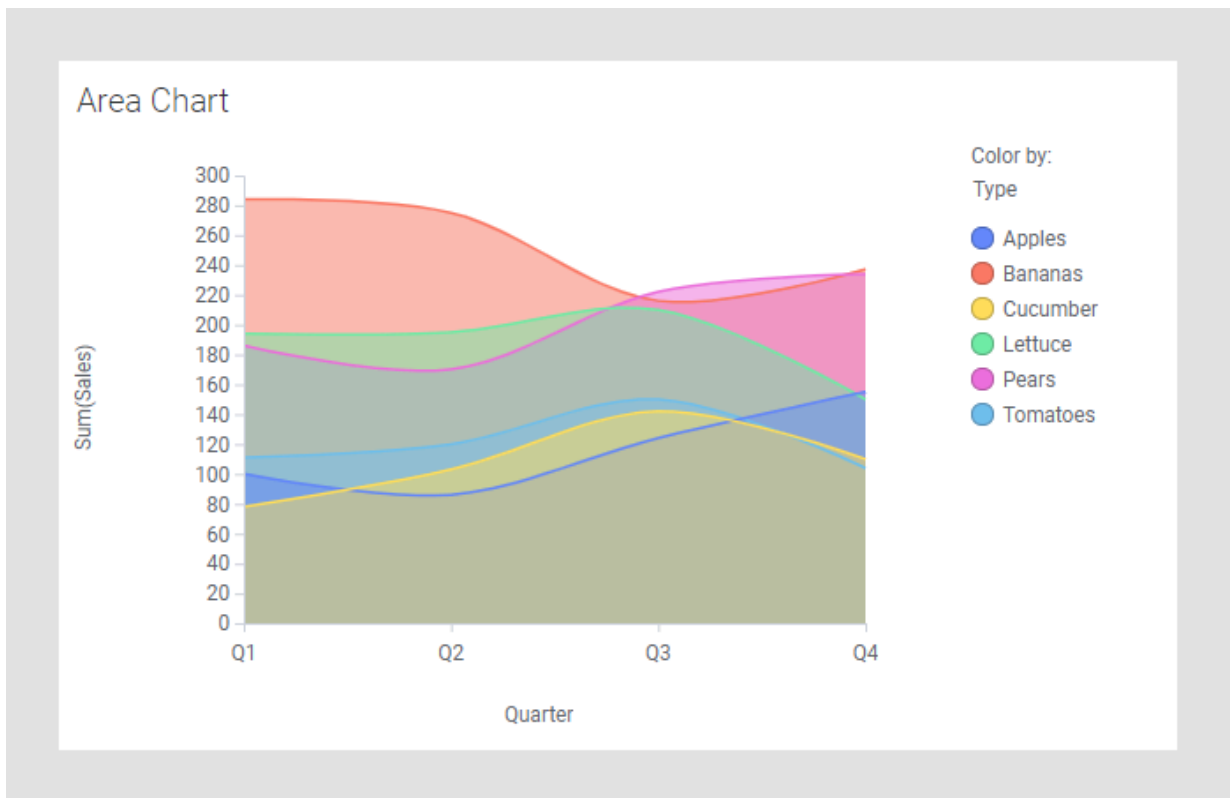
The Mods development tool is where you can connect to the mods you are developing. You can reach the tool from **Tools > Development > Mods development**.



## Visualization mods

Although Spotfire offers many different visualization types, you might miss a certain way to visualize your data. To fill that gap, Spotfire provides a framework which makes it possible for a developer to extend Spotfire with new customized visualizations. Through the provided Spotfire mod API, these customized visualizations can be created using JavaScript or TypeScript, and they are called visualization mods.

The area chart below is an example of a visualization mod that has been created using this framework.



### How does a visualization mod work?

You can think of a visualization mod as a visualization, whose appearance is specified by the mod developer who created the visualization mod, but where all data-related functionality is handled in the same way as in any of the native Spotfire visualizations. An end user feels that the visualization mod is integrated with Spotfire, because the visualization responds to interactions in the same way as native visualizations do. For example, it is possible to drag a visualization mod to the visualization canvas, and change what is selected on the various visualization axes. Moreover, filtering of the data works as usual across all visualizations, no matter if they are native visualizations or visualization mods. The end user might not even notice there is a difference.

The visualization mods can be based on any of the data sources supported by Spotfire; in-memory data, in-database data, streaming data, and data-on-demand.

### Sharing visualization mods with others

A visualization mod in Spotfire can be made accessible to other users in various ways:

- It can be saved to the Spotfire library.

When the visualization mod has been saved to the library, it can be added to analyses and also pinned to the visualization flyout for easy access. Users can browse and search the library for visualization mods.

- It can be embedded in an analysis.

The analysis can then be saved to the library, or saved as a local file.

## Developing visualization mods

A developer of a visualization mod needs a running instance of a Spotfire client, and a source code editor. [Examples](#) of visualization mods, which can serve as starting point for developers, are available for download from Spotfire Mods on GitHub. The examples to download are built using the Visual Studio Code editor. When using Visual Studio Code as editor, it is possible to get a live preview of the mod within the Spotfire client while developing.

For more information about the actual development and the tools, see [the Spotfire® Mods website](#) on GitHub.

## Version handling

By default, if a visualization mod is resaved to the Spotfire library, all instances of the visualization mod are updated in all analyses where it is used.

See also [Manage trust](#).

## Action mods

An action mod is a collection of actions (scripts) that are packaged together, which makes it easy to share and reuse actions in your analyses. Actions within a mod can be run from triggers in visualizations or text areas, or directly from the library. The action mod is developed in a similar way as a visualization mod; it can be developed and tested within the context of an analysis where it will be used. The scripts in an action mod can share common code and include libraries needed by the scripts, and they can be trusted and saved for reuse in multiple analyses.

With action mods, you run scripts in a sandboxed environment so that a script can only affect yourself, and you can only access your own data. This way, everybody with the required licenses can run scripts that they have authored themselves, because the script cannot affect the other users, read other users' data or affect the system as a whole. It is also possible to configure analyses that let any user run a script created by another user that they trust.



Some scripts might still be able to, for example, corrupt a Spotfire analysis or leak sensitive data that the analysis fetches from a data source so always use your judgment when you decide to run a script.

## Sharing action mods with others

An action mod in Spotfire can be made accessible to other users in various ways:

- It can be saved to the Spotfire library.

When the action mod has been saved to the library, actions within that mod can be run ad hoc from the library, and it can also be pinned to the Actions flyout for easy access. Users can browse and search the library for action mods.

- It can be embedded in an analysis.

The analysis can then be saved to the library, or saved as a local file.

## Developing action mods

A developer of an action mod needs a running instance of a Spotfire client, and a source code editor. For more information about the actual development and the tools, see [the Spotfire® Mods website](#) on GitHub.

## Version handling

By default, if an action mod is resaved to the Spotfire library, all instances of the mod are updated in all analyses where it is used.

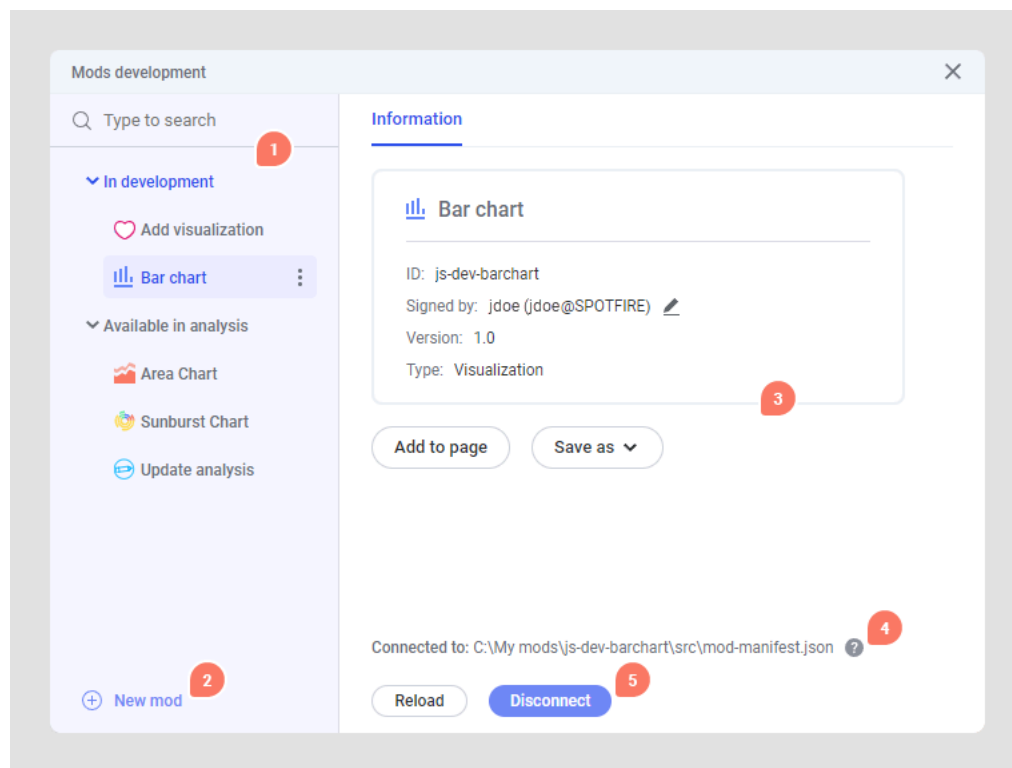
See also [Manage trust](#).

## The Mods development tool

The Mods development tool gives you an overview of the mods that you have in development and that are used in the analysis. You can also test run actions in your action mods and save configurations for later use.

To open the tool, select **Tools > Development > Mods development**.

### Information tab

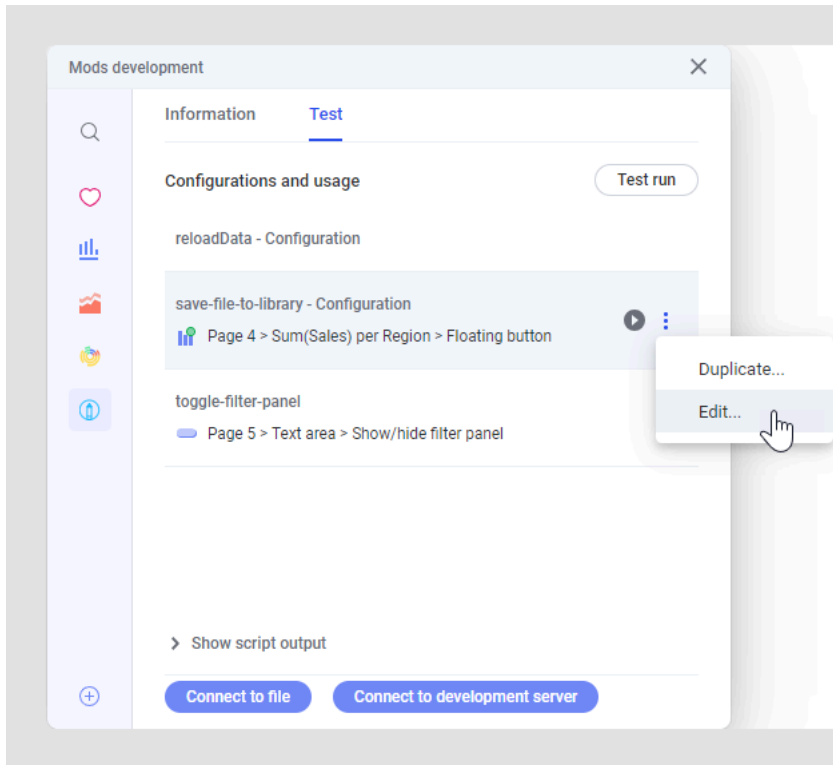


1. In the sidebar to the left, you can see which mods are in development, and which ones are available in the analysis. To view more options, open the menu that is shown to the right when you hover over a mod in the sidebar. You can, for example, save the mod to the library or locally, or remove the mod from the analysis. Which options are available depends on whether the mod is in development or not. The sidebar to the left is always available, but can be collapsed to save space.
2. Click **New mod** at the bottom of the sidebar to connect to a new mod.
3. In the area to the right, you find information about the selected mod, and you can save the mod. For visualization mods, you can add the visualization to the active page.
4. When a mod is in development, you can see how you are currently connected to the mod; either a path to the folder containing the manifest file, or to the development server.
5. When the mod is in development, as in the image above, two buttons will be available at the bottom of the tool: **Reload** and **Disconnect**. When the mod is not in development, you will see the buttons **Connect to file** and **Connect to development server** if you are working in the installed client. In the

web client, you can only connect to a mod via a development server, which means only one button will be shown.

## Test tab

Action mods also have a **Test** tab where you can see which actions in the mod currently have one or more configurations. You can also see which configurations are used in the analysis and have been added to any triggers.



## Test running an action in an action mod

When you are developing an action mod, it can be useful to perform test runs of actions along the way, to make sure the scripts work as intended. When doing test runs of your actions, you can save configurations of actions containing parameters so that those actions can be run again with the same configuration, or be used later on, for example on an action trigger in a visualization or an action control in a text area.

### Prerequisites

You must have the Develop Mods license feature and the analysis must be in Editing mode. The action mod containing the action you want to test run must either be available in the analysis or in development. For more information, see [Developing mods](#) on page 630.

### Procedure

1. On the menu bar, select **Tools > Development > Mods development** to open the Mods development tool.
2. In the sidebar to the left, click the mod containing the action you want to test run, and then open the **Test** tab.
3. Click **Test run**.

4. In the flyout that opens, select the action you want to run and click **Next**, or **Run** if the action does not have any parameters.
5. If the selected action has parameters, provide the necessary input in the flyout that opens, and then click **Run**. For actions with parameters, you can enter a name for the configuration to save it with the specified input for the parameters. See [Configuring action parameters](#) to learn more about the available options.

The action is run. You can view existing configurations on the **Test** tab of the Mods development tool as well as in the details view of the [Actions flyout](#).



Enter a descriptive name for the configuration if you intend to use it later on.

## Actions

Actions is a Spotfire concept that offers a wide range of possibilities to perform tasks from within an analysis, and can also include tasks that affect external systems. The actions can be added to text areas and visualizations, and they are initiated using various controls or triggers. For example, you can create actions that control the workflow through an analysis, run scripts, or let people send data externally. That is, the contents of the actions can differ a lot, and in addition, the way they are added differs depending on the context.



Addition of text areas and some types of actions are handled using the installed client.

### Actions that perform tasks within an analysis

These actions can be triggered from text areas or from some visualizations. Text area buttons and links are provided that can initiate tasks like application of bookmarks, navigation workflows, or execution of data functions and scripts. These types of actions can also be defined in KPI charts and graphical tables. You can also add triggers performing certain actions in some other visualizations. Scripts can be added as a part of an analysis. If the scripts are packaged together as actions in an [action mod](#), they can be signed and shared as a part of the mods framework. Script actions can be added to links, images or buttons in text areas, they can be triggered when a document property is changed, or they can be triggered when you are clicking on a graphical item in a visualization.

For more information on how to create such actions, see [Adding actions to a text area](#) on page 636, [Adding an action trigger to a visualization](#) on page 649, and [Adding actions performed when clicking on visualization items](#) on page 646.

### Actions that perform tasks involving external systems

Your findings in a visualization might sometimes make you want to take an action in another system. For example, you might want to send certain information and data that you see to another application. Such external actions can be triggered directly from a visualization. A configurator defines which triggers should be available in the visualizations, and what information the action should send. Anyone opening the analysis can potentially use the configured action trigger to send data. Before you give approval to send any data externally, you can [review it](#).

See [Adding a new external action to an analysis](#) on page 650 for more information about external actions.

## Adding actions to a text area

As an analyst author (and a script author), you can decide to add different types of actions to buttons, links or images in a text area (using the installed client). The action controls can open Spotfire tools that work on the range of filtered or marked data, apply bookmarks, or navigate to a certain page or



visualization in the analysis. It is also possible to add your own custom actions using scripts, to refresh data function calculations, or to export prepared reports to PDF documents.





Text areas must be authored in the installed client.

### Prerequisites

You must have some data loaded in the analysis (in the installed client) and the analysis must be in **Editing** mode.

### Procedure

1. On the text area title bar, click **Edit Text Area** .
2. In the Edit Text Area dialog, click **Insert Action Control** .
3. In the Action Control dialog, in the panel to the left, decide which type of action to add: **Actions**, **Export Report**, **Script** or **Data function**.  
For more information about the different options, see the examples in the following topics.
4. Type a **Display text** to be shown in the text area.
5. Select a **Control type**: Link, Button or Image.  
If you choose to add an action to a clickable image, you must also browse to add the image of your choice.
6. When you are done, click **OK** and save the text area.

### Result

The action control is added to the text area.

## Adding actions linked to functions, bookmarks, pages or visualizations

Using the installed client, you can add actions that include showing or hiding panels, performing various operations on filtered or marked rows, applying bookmarks, switching pages or navigating to visualizations. Multiple actions of this type can be performed in just one click.

### Prerequisites

You must have some data loaded in the analysis (in the installed client) and the analysis must be in **Editing** mode.

Follow the steps in [Adding actions to a text area](#) on page 636 or [Adding actions performed when clicking on visualization items](#) on page 646 and make sure you select **Actions** in step 3.

### Procedure

1. In the **Available actions** list, select the action that you want to perform.  
All actions that can be included in a multiple action link are listed, grouped by type. Click the plus sign to expand a group. Click an action (a function, a bookmark or a page) to select it.

#### Comments about Reset All Visible Filters

This functionality is based on which filters are shown in the Filters panel only, and not related to any filters that you have added to a text area. This means that if a filter is hidden from the Filters

panel, it will not be updated using this action, regardless if you can see it in the text area or not. The Filters panel does not have to be shown for the action to work, though.

### Comments about bookmarks

For bookmark components, the active visualization always takes precedence over the active page.

The filter settings of the active filtering scheme are changed, including the setting determining how related data tables are affected by each other's filtering.

The markings component of a bookmark takes the analysis back to the state all markings had when the bookmark part was captured. If a marking has been removed, that marking is simply ignored.

### Comments about marked rows

The marking applied when adding any of the Marked Rows functions is the one used by the active data table. The active data table is the one used by the visualization that was active before clicking on the text area. If the marking for the active visualization is set to (None), nothing will happen when clicking on the link.

2. Click **Add**.
3. If you want to include more steps, repeat steps 1-2.

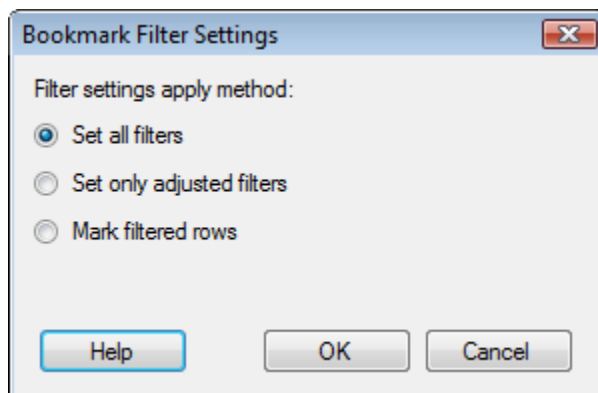


Some types of actions allow you to specify additional settings by clicking the **Settings** button. For example, you can change filter settings for a bookmark or toggle the visibility of the Filters panel or the Details-on-Demand.

4. When you are done, click **OK**.

## Bookmark filter settings

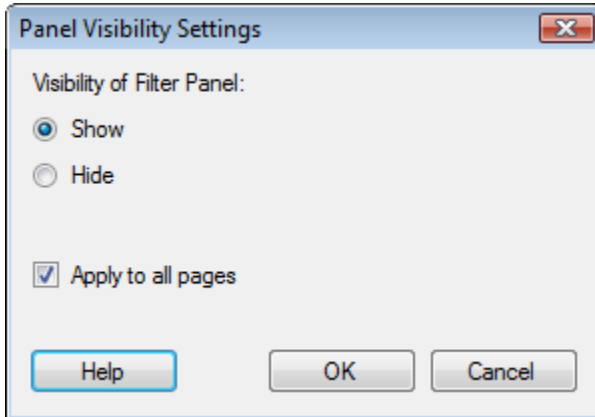
In the Bookmark Filter Settings dialog you can change how to apply the filtering from a bookmark.



Option	Description
Filter settings apply method	
Set all filters	Applies the exact filter settings for the active filtering scheme, as stored in the bookmark.
Set only adjusted filters	Keeps everything as it is in the filters panel except for those filters that were changed in the bookmark, which are updated.
Mark filtered rows	Uses the filter condition from the bookmark to mark rows in the analysis, but leaves the filtering exactly as it was before applying the bookmark.

## Panel visibility settings

In the Panel Visibility Settings dialog you can specify the visibility of the selected panel, and determine whether to show it on all pages or on the active page only.



Option	Description
Visibility of [Panel Name]	
Show	The selected panel is shown when clicking the action link or button.
Hide	The selected panel is hidden when clicking the action link or button.
Apply to all pages	Select the check box to apply the change to all pages. Clear the check box to only change the visibility on the active page.

## Adding a script action to a text area or visualization item

As a Spotfire Analyst user (and a script author), you can define your own scripts and add them to action controls in a text area, or as an on-click action for certain visualization items, using the installed client. By using scripts included in an action mod, analysts who do not create scripts themselves can add shared scripts developed by trusted signers. The scripts allow you to incorporate simple actions or tools into your analysis without having to deploy complete AddIns to the server. The scripts have full access to the Spotfire API.

See [Developing mods](#) for more information about action mods.

For security reasons, there are some restrictions for using scripts. For more information, see [Usage of scripts and data functions](#) on page 678. Script authors should also make sure to save trusted scripts to the library, so that other users can benefit from their work. For even better shareability, scripts can be included in a signed action mod. See [Manage trust](#) on page 680 for more information.

See also [IronPython Scripting in Spotfire](#) on the Community for more examples on using scripts in Spotfire.

In HTML mode it is also possible to add some functionality using JavaScript by editing the HTML of the text area. See [JavaScript example scripts](#) on page 644 for more information.



On the [Community](#), the article [Best Practices for Writing Custom JavaScript Code in Text Areas](#) gives an introductory overview of how you can use JavaScript in text areas in Spotfire.



Text areas must be authored in the installed client.

## Prerequisites

You must have some data loaded in the analysis (in the installed client) and the analysis must be in **Editing** mode.

Follow the steps in [Adding actions to a text area](#) on page 636 or [Adding actions performed when clicking on visualization items](#) on page 646 and make sure you select **Script** in step 3.

## Procedure

1. In the Action Control or Action Settings dialog, under **Available scripts**, click the script of interest. If you do not have any scripts available in the analysis, you can **Add** existing scripts from an action mod file (\*.mod), from an action mod in the library, or add a script that has previously been saved in an analysis file, either locally or in the library. You can also define a new script, if you have the required permissions. For tips on how to define a new script, see *Example script workflows* below.



Spotfire compares the scripts you are trying to import with already existing scripts. A script whose script body is identical to an existing script will not be imported even if it has a different name.

The **Remark** field can contain information about whether a script is trusted or not. Scripts that you have created yourself are trusted, and so are scripts saved as trusted to the library by a member in the Script Author group. Scripts from an action mod can be either trusted or not trusted, depending on whether you, or the administrator, have trusted the mod or the signer in advance.

If you receive an analysis file containing a script directly from another person, the script will be listed as "Not trusted". This does not necessarily mean that the script contains any unsafe material, but you should determine whether you trust the script by inspecting it in the Edit Script dialog. By clicking **OK** in the Edit Script dialog, the script will be trusted. If you know all scripts are safe, you can go to [Manage trust](#) and click **Trust all**.

However, due to security reasons, scripts cannot be used in a TIBCO Cloud Spotfire web client environment unless they come from trusted action mods. In a Spotfire on premises web client environment, scripts can be executed if they come from trusted action mods or are saved as trusted by a member in the Script Author group.

2. Click each parameter (if there are any) in the **Necessary input for the selected script** field and specify the **Input for the selected parameter**. If the script needs some kind of input values, then these must be specified before the script can be executed. Different parameter types have different input settings.

For value parameters, you can choose whether to type a text-based or numeric **Value** directly, to associate the selected parameter with a **Property**, or to calculate the value via an **Expression**. Other parameters might need a specification of a visualization, a page, or a data table to work with.

3. When all script parameters have been defined, click **OK**.
4. If you are adding an action to a text area, click **Save** when you are finished.
5. If you are adding an action to a text area, click **Close** in the upper-right corner to exit the Edit Text Area dialog.

## Example script workflows

Below are more details about how to create and use a few example scripts in Spotfire. These steps can also serve as instructions for how you should work when creating your own scripts. See also [IronPython example scripts](#) on page 643.

## Creating the "Change visualization title" script

This example shows how you can add a script that changes the title of a visualization.

See [Adding actions to a text area](#) on page 636 for information about how to reach the Action Control dialog.

### Procedure

1. In the Action Control dialog, click **Add > New script**.
2. In the New Script dialog, type the **Script name** `Change visualization title`.
3. In the **Script** field, type or paste the script:
 

```
visual.Title = title
```
4. Click **Add**.
5. In the Add Script Parameter dialog, enter `title` as the **Name**, and under **Type**, select **String**.  
The title must be a string which can be defined either by a simple text value, a string property or an expression.
6. If desired, add a **Debug value** which can be used for testing the script.
7. Click **OK**.  
The "title" parameter is added to the Script parameters list.
8. Click **Add** to define the second parameter, and in the Add Script Parameter dialog, enter `visual` as the **Name**, and under **Type**, select **Visualization**.  
This parameter is used to point out which visualization to change the title on.
9. Select a **Debug value** visualization to use when testing the script.
10. Click **OK**.  
The "visual" parameter is added to the Script parameters list.
11. If desired, test the script by clicking **Run Script**.  
The selected visualization title is updated with the provided debug value. Click **Undo** in the New Script dialog to revert to the old title.
12. Click **OK**.

### Result

The "Change visualization title" script is added to the **Available scripts** list in the Action Control dialog and the defined parameters are shown in the **Necessary input for the selected script** list (as long as the script is selected in the Available scripts list).



To instead change the title of the active visualization, use the following script:


```
Document.ActiveVisualReference.Title = title
```

## Using the "Change visualization title" script in a text area

When you use a script in an action control you can define how to provide the necessary input for all parameters of the script. This example shows how you can use the script from the previous example on a button in a text area.

### Procedure

1. Follow the step instructions above and make sure the "Change visualization title" script is selected in the Action Control dialog.

2. In the Action Control dialog, under **Necessary input for the selected script**, click the parameter **title**.
3. Under **Input for the selected parameter**, you determine whether to define the title using a predefined Value, a Property or an Expression. In this example, click **Property** to be able to change the title using a property control.
4. Click **Select Property**.
5. In the Select Property dialog, on the **Document Properties** tab, click **New**.
6. In the New Property dialog, type a suitable **Property name**, for example, `visualization.title`.
7. From the **Data type** drop-down list, select **String**.
8. Type a **Value** to be the first (default) value of the property. For example, `My Visualization Title`.
9. Click **OK** to close all dialogs.  
The action control is added to the text area.
10. Click **Save**  when you are done editing the text area, and then close the Edit Text Area dialog.
11. Test the action control by clicking it in the text area.



The property controlling the visualization title could in turn be modified using a property control (for example, an input field) in the text area. See [Adding a property control to a text area](#) on page 325 and [Using document, data table or column properties in an analysis](#) on page 250 for examples of how you can add property controls.

My Visualization Title

Change Title

## Creating the "Change filtering scheme" script

This example shows how you can add a script that changes the filtering scheme.

See [Adding actions to a text area](#) on page 636

### Procedure

1. In the Action Control dialog, click **Add > New script**.
2. In the New Script dialog, type the **Script name** `Change filtering scheme`.
3. In the **Script** field, type or paste the script:

```
from Spotfire.Dxp.Application.Filters import FilterPanel

# Get hold of the filters panel for the current page.
panels = Document.ActivePageReference.Panels
# The return value and out parameters are returned as a tuple
# in IronPython and can be bound with pattern matching.
(found, filterPanel) = panels.TryGetPanel[FilterPanel]()

# Find the filtering selection named "Filters2"
# and set it as current.
for filteringScheme in Document.FilteringSchemes:
    filteringSelection = filteringScheme.FilteringSelectionReference
    if filteringSelection.Name == "Filters2":
        filterPanel.FilteringSchemeReference = filteringScheme
```


The included example script expects to find a filtering scheme called "Filters2" and changes the filtering scheme of the active page to the Filters2 filtering scheme.

4. Click **OK** to close the New Script dialog.

## Using the "Change filtering scheme" script in a text area

This example shows how you can use the script created in the previous example on a button in a text area.

### Procedure

1. Make sure that the "Change filtering scheme" script is selected in the Action Control dialog.
2. Click OK.  
This script requires no input from the end users, but if there is no "Filters2" filtering scheme in the analysis, then nothing will happen when the action control is clicked.
3. Click **Save**  when you are finished editing the text area, and then close the Edit Text Area dialog.
4. Test the action control by clicking it in the text area.



## IronPython example scripts

This topic lists a number of example scripts that can be executed by clicking on action links or buttons in the text area.

See also [Adding a script action to a text area or visualization item](#) on page 639 for more examples and detailed instructions on how to add scripts to a text area, and see [IronPython Scripting in Spotfire](#) on the [Community](#) for many more examples, as well as tutorials, on using scripts in Spotfire.



If the expression parameter is tied to a string property value, you can use a property control (for example, a drop-down list) with a number of predefined expression alternatives to create an analysis where complex expressions can be assigned to visualization axes in an easy way. See [Adding a property control to a text area](#) on page 325 for information about how to add property controls.

### Refresh of calculations, on-demand data and data functions

```
# Example script that refreshes a table driven by
# a calculation, a data function
# or an information link loaded on demand.
# The script takes a parameter "table" of type DataTable
if table.IsRefreshable and table.NeedsRefresh:
    table.Refresh()
```

### Add columns from an SBDF file to a data table in the current document

```
# This script adds columns from an sbdf file to a data
table
# in the current document.
#
# Four arguments are expected:
# table - The data table in the current document
# that columns will be added to.
# path - The path to the sbdf file with columns to add.
# ColumnNameInTable - A column in the data table.
# ColumnNameInFile - A matching column in the sbdf file.
# The columns are used to join new data to the current data.
from Spotfire.Dxp.Data.Import import SbdfFileDataSource
from Spotfire.Dxp.Data import AddColumnsSettings, JoinType,
DataColumnSignature, DataType
# Create the join condition map. The column with the name
# specified in ColumnNameInTable from the current data table
# will be joined with ColumnNameInFile in the sbdf file.
columnInTable =
DataColumnSignature(table.Columns[ColumnNameInTable])
```

```
columnInFile = DataColumnSignature(ColumnInFile,
    DataType.String)
joinConditionMap = {columnInTable : columnInFile}
ignoredCols = [ ]
settings = AddColumnsSettings( joinConditionMap,
    JoinType.InnerJoin, ignoredCols)
ds = SbdFileDataSource(path)
table.AddColumns(ds, settings)
```

## Use a specific visualization ("Visual") as input and then change the X-axis of the visualization

```
# Sets the value of the XAxis Expression of a visual.
# This script expects two arguments:
# visual - The visual to the set the X-axis expression on.
# expression - The expression to set.
from Spotfire.Dxp.Application.Visuals import VisualContent
# Get the content of the visual. Use the most general type
# so that the script works for all VisualContent classes that
# have an X-axis property.
vc = visual.As[VisualContent]()
vc.XAxis.Expression = expression
```

## JavaScript example scripts

This topic shows some examples of scripts based on JavaScript that can be used in the text area to provide some simple interactivity.



On the [Community](#), the article [Best Practices for Writing Custom JavaScript Code in Text Areas](#) gives an introductory overview of how you can use JavaScript in text areas in Spotfire.



The JavaScript you add to the text area will execute in the same environment as the JavaScript defined by Spotfire to implement all functionality in the text area and other Spotfire visualizations. This means that there are a lot of JavaScript libraries defined by Spotfire and third parties in scope, including jQuery (\$) and jQueryUI. Spotfire provides no guarantee whatsoever of the correctness, usability or compatibility of the libraries in scope. Future versions of Spotfire are likely to include other, upgraded and or incompatible libraries. Libraries can also be removed in future versions. Therefore, if you want to use libraries like jQuery or jQueryUI you must import those libraries yourself instead of relying on the versions used by Spotfire. See the Community article [How to include your own instances of jQuery and jQueryUI in Text Areas](#) for more information.

## Change the background color of an element when the mouse is over it

The following script uses two parameters, "id" and "color", where "id" is the id of the element to change background color on, in the example set to "my-p", and "color" is the background color to set (for example, #e7e3e7). Both parameters are specified as String parameters in the Insert JavaScript dialog.

```
-----
var elem = document.getElementById(id);
if (!elem)
{
    return;
}

var oldBgColor = elem.style.backgroundColor;
var onEnter = function()
{
    elem.style.backgroundColor = color;
};

var onLeave = function()
{
    elem.style.backgroundColor = oldBgColor;
};
```



```
elem.onmouseover = onEnter;
elem.onmouseout = onLeave;
```

The script is then used by calling the id in the text area HTML.

For example, by adding the required id to a paragraph:

```
<p id="my-p">A JavaScript changes the background color when the mouse is over this
paragraph.</p>
```

This will result in a text area paragraph where the background color changes to the specified color when you move the mouse over the text in the paragraph.

### Signal when the text area is ready to be rendered to PDF or image

If your script contains asynchronous calls, you must use the `SF.setBusy()` API to make sure the text area will render nicely when exported to PDF or image.

By executing

```
-----
SF.setBusy(true);
-----
```

the export browser will not render a PDF/image until the script executes:

```
-----
SF.setBusy(false);
-----
```

### Adding an action to export a report to a text area

As an analyst author, you can add a button, link or image in a text area (using the installed client) that exports a report to PDF when clicked.

#### Prerequisites

Follow the steps in [Adding actions to a text area](#) on page 636 and make sure you select **Export Report** in step 3.

#### Procedure

1. Under **Available reports**, locate the report of interest.  
If you do not have any reports available in the analysis, you must first create at least one report as described under [Preparing a report](#) on page 945.
2. When you are done, click **OK**.

### Adding an action that refreshes a data function to a text area or a visualization item

As an analyst author, you can add a button, link or image in a text area (using the installed client) that refreshes a previously executed data function when clicked.



For security reasons, there are some restrictions for using data functions. For more information, see [Usage of scripts and data functions](#) on page 678 and *Trust for script written in IronPython, JavaScript or within data functions* under [Manage trust](#) on page 680.

## Prerequisites

You must have some data loaded in the analysis (in the installed client) and the analysis must be in **Editing** mode.

Follow the steps in [Adding actions to a text area](#) on page 636 or [Adding actions performed when clicking on visualization items](#) on page 646 and make sure you select **Data function** in step 3.

## Procedure

1. Under Available data functions, locate the data function of interest.



Only the data function that have been executed within the current analysis and that do not use the Refresh function automatically feature are listed here. Read more about data functions under [Data functions and the f\(x\) flyout](#) on page 662.

2. When you are done, click **OK**.

## Adding actions performed when clicking on visualization items

Actions can be defined so that something happens when you click a dynamic item in a graphical table or a text area, or when clicking a tile in a KPI chart. The actions can open Spotfire tools that work on the range of filtered or marked data, apply bookmarks, or navigate to a certain page or visualization in the analysis; the same things that are available for action controls in a text area. It is also possible to add your own custom actions using scripts, or to refresh data function calculations.



See [IronPython Scripting in Spotfire](#) on the [Community](#) for more examples, as well as tutorials, on using scripts in Spotfire.

## Prerequisites

Adding actions can only be done using the installed client.

## Procedure

1. In the visualization properties/settings for the visualization item (sparkline, calculated value, icon, bullet graph or KPI) to update, locate **Actions**.
2. Select **Perform action on click**.
3. Click **Settings**.

4. In the Action Settings dialog, select the type of action to add: **Actions**, **Script** or **Data Function**.  
For more information about the different options, see the examples in [Adding actions linked to functions, bookmarks, pages or visualizations](#) on page 637, [Adding a script action to a text area or visualization item](#) on page 639 and [Adding an action that refreshes a data function to a text area or a visualization item](#) on page 645.

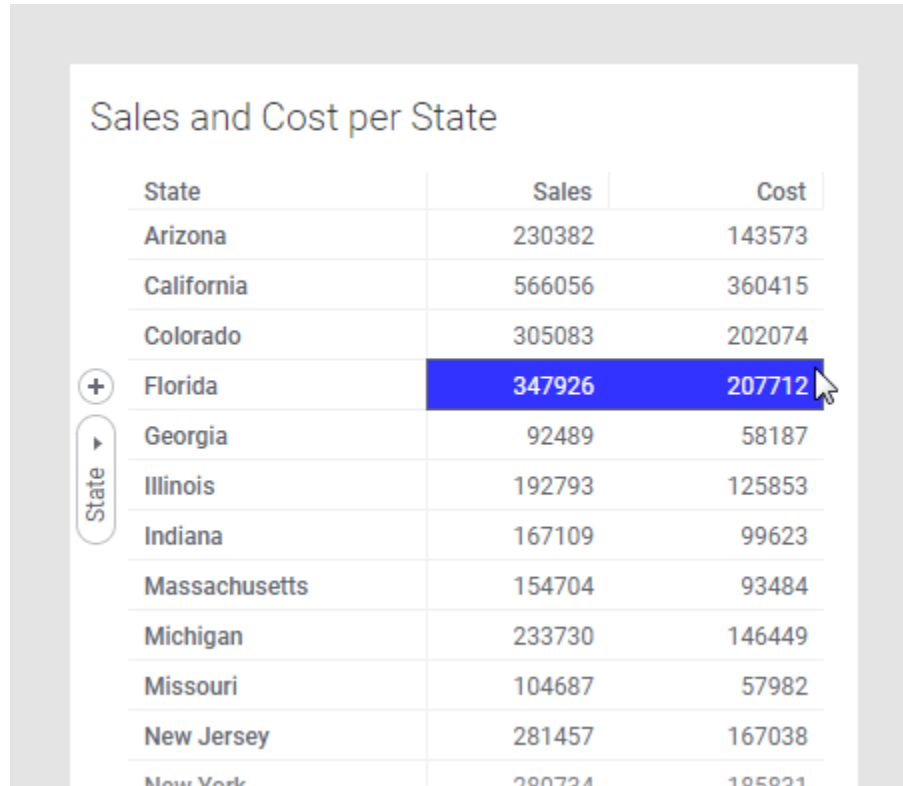
5. Type a **Description** for the action.

This text can be shown in the tooltip when hovering with the mouse pointer over the item connected to the action. By typing a good description you can help other users understand what will happen when clicking on the item. For example, you can write "Go to page 2", "Refresh the calculation", or similar.

## Configure details visualization

Below is an example of what you can do by setting up an IronPython script action on two dynamic items of the type calculated value in a graphical table.

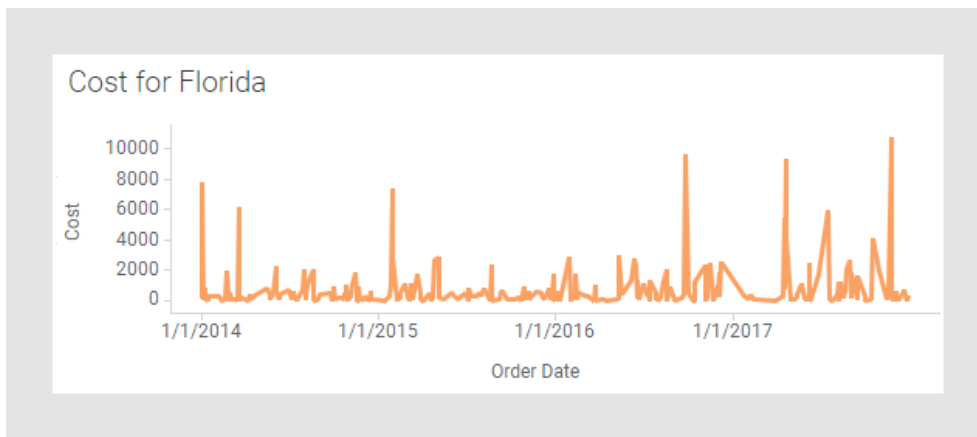
This example uses a fictive data table containing sales and cost for a number of products in the different states of USA. The graphical table has been configured to show all states on the Rows axis and two dynamic items of the type calculated value have been added, showing the sum of sales and the sum of cost, respectively:



State	Sales	Cost
Arizona	230382	143573
California	566056	360415
Colorado	305083	202074
Florida	347926	207712
Georgia	92489	58187
Illinois	192793	125853
Indiana	167109	99623
Massachusetts	154704	93484
Michigan	233730	146449
Missouri	104687	57982
New Jersey	281457	167038
New York	280731	185831

Another visualization, a line chart, has been configured to show either the sum of sales or the sum of cost for the marked state in the graphical table over the time, thus displaying the details behind the total sum for each state. In contrast to a standard details visualization controlled by the marking, this visualization will show different data depending on which cell you click on; clicking in the Sales column will show the sales for that state and clicking in the Cost column will show the cost.

This is done using an action script on the columns in the graphical table which defines what to show on the Y-axis and automatically limits the line chart by a boolean expression set to show data for the state of the clicked row in the graphical table only. The title of the details visualization is also updated by the same script:



1. From the visualization properties, open the Calculated Value Settings dialog for the first column, "Sales", (on the Axes page) and go to the **Actions** section.
2. Select **Perform action on click**.
3. Click **Settings**.
4. Click **Script**.
5. Click **Add > New script**.
6. In the New Script dialog, type a **Script name** (for example, "Configure details visualization").
7. Type a **Description** (for example, "Configures a visualization to show details of a graphical table cell.")
8. Copy the script below and paste it into the **Script** field.



The example script assumes that a column called "State" is available in the data table.

9. Next to Script parameters, click **Add**.
10. In the Add Script Parameter dialog, type the **Name** detailsVis, as expected by the script.
11. Let the **Type** be a **Visualization** and select the line chart to use as the details visualization from the list.
12. Click **OK** in all dialogs to close them.
13. Add the defined script to the other column as well, to be able to show either Sales or Cost details in the line chart.

```
# This script is intended to be run as an action on a
# Calculated Value miniature visualization in a graphical
# table.
#
# It expects one Visualization argument, detailsVis, and
# it configures the specified visualization to show details
# for the clicked cell.
#
# When executed as an action in a graphical table,
# the variable Context is bound to an instance of
# MiniatureVisualizationActionContext (see API documentation).

from Spotfire.Dxp.Application.Visuals import VisualContent

# This script assumes that the row axis of the Graphical Table
# is configured with one column: State.
#
# Get the miniature visualization and the value of the row
# axis hierarchy (the state column) for the clicked cell:
```

```

clickedMiniVis = Context.Visualization
state = Context.HierarchyPathValues[0]

# Get the content of the visual that shall be configured to
# show details. Use the most general type so that the script
# works for all VisualContent classes that have a Y-axis
# property. This works well for a Line Chart, for instance.

vc = detailsVis.As[VisualContent]()

# Configure the Title and YAxis of the details visualization:

vc.YAxis.Expression = clickedMiniVis.ValueAxis.Expression
detailsVis.Title = clickedMiniVis.Title + " for " + state

# Limit the data of the details visualization to only use
# data for the selected state:

vc.Data.WhereClauseExpression = "State = \" " + state + "\" "

```

## Adding an action trigger to a visualization

You can add a trigger to a visualization and connect it to an action that you want to run when clicking the trigger. You can select external actions or actions from action mods.

### Prerequisites

You must have some data loaded in the analysis and a visualization where you want to add a trigger. The analysis must be in **Editing** mode. If you are working in the web client, make sure the action you want to use is available in the **Actions** flyout.

### Procedure

1. Select the visualization you want to add an action trigger to.
2. Open the **Visualization properties** panel.
3. Click **Add** and select **Action trigger** from the menu.  
An unconfigured **Action trigger** card is added to the panel.
4. Select which trigger type you want to use.
5. Click **Select action**.  
The **Actions** flyout is opened.
6. Locate the action you want to add to the trigger, and select it by clicking the plus sign to the right. Action mods can contain several scripts, and you might need to expand the item to select a specific action to add to the trigger. In the installed client, you can click **More actions** to open a menu where you can locate action mods in the library or locally, or add an external action. See [Adding a new external action to an analysis](#) to learn more about the option **Add action from external service**.



Right-click an item and select **View details** to see if any configurations exist for the action already. You can also see if the configurations are used somewhere in the analysis.

7. If the selected action requires input, provide the necessary input in the flyout that is opened, and then click **Done**. See [Configuring action parameters](#) to learn more about the available options. The action trigger is added to the visualization with the selected action.

## Adding a new external action to an analysis

Actions that include sending data to external systems make use of integration and connection services in TCI. Services can be created for specific action needs. From Spotfire, the services are then accessed using TIBCO Cloud Mesh, a global repository of these kinds of services.

### Prerequisites



To add or use external actions, you need TIBCO Cloud™ Integration license (TCI). For more information, see [Enabling external actions with TIBCO Cloud™ Integration](#) on page 651 .

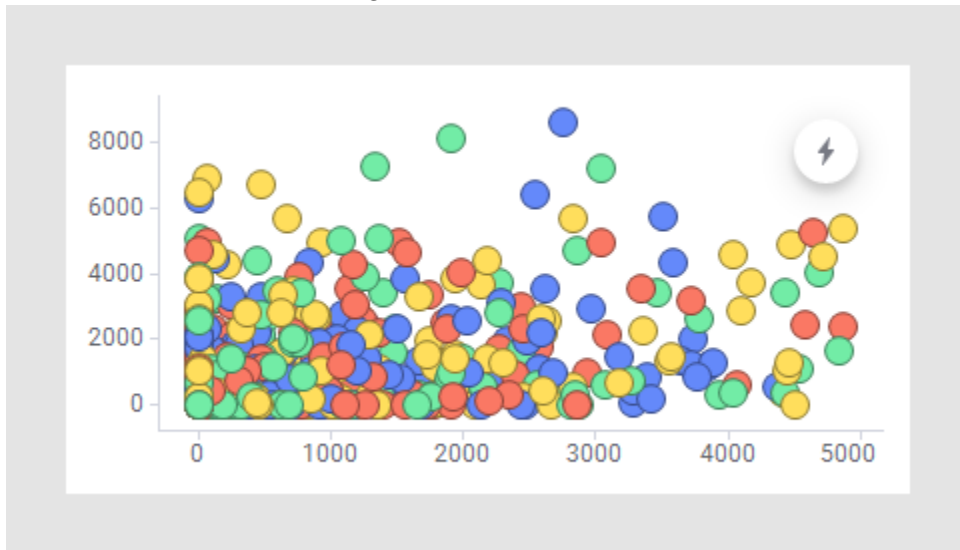


External actions must be added to the analysis using the installed client.

See [Actions](#) for a short description of different types of actions.

An external service can contain various parameters to configure. When adding an action that calls a service, any defined parameters will be provided automatically in a configuration flyout in Spotfire, where you map them to your data. For example, you can define what kind of data in the visualization to send to the external system.

A configured action is then activated using some type of trigger in the visualization, like a floating button, as shown below, or a right-click menu item.



When adding a new action, you add it to a visualization of your choice, and you can specify one or more triggers in that visualization. When you have added an action, it becomes available for selection in all visualizations in the analysis. That is, an action can be added to more than one visualization in an analysis, using the same type of trigger or a different one. You can also add more than one action to a single visualization.

### Procedure

1. On the authoring bar, click **Actions** ⚡ to open the flyout.
2. Click **More actions** and select **Add action from external service**.
3. In the TIBCO Cloud Login dialog, log in to TIBCO Cloud, and follow the steps.
4. In the Select a service from TIBCO Cloud Mesh dialog, locate and select the service to use for the action.  
The **Configure action** flyout is opened, listing the parameters that are defined for the service.

5. Optional: Click the pen to rename the action, and type a description.  
A description of the action is recommended to inform people who run it about the purpose of the action.
6. Click ⚡ to select an icon that will represent the action.
7. Expand the parameters, and configure the action, that is, map inputs to the contents of your analysis. For more details on the available options for the parameters, see [Configuring action parameters](#).

It is the configurator of the action who determines what data the action will send. The configurator might select a data table, or one or more columns, to send data from, add a document property (perhaps a property that people can change using controls in a text area), or even type a value directly. By specifying the type **Prompt for input when running action**, the person who runs the action will be asked to provide the input. Optionally, a description can be added to a parameter using prompt as input type. A question mark icon will then be shown next to that parameter in the prompt, and the description will be available in the tooltip.

8. Click **Next**.  
The **Review action configuration** flyout is opened. The flyout shows the configuration and affected data that potentially will be sent externally.
9. Make sure the configuration is correct, and click **Done**.

## Result

You are now the signer of the action. The action is added to the **Actions** flyout, and will also be available for selection when adding action triggers in visualizations. See [Adding an action trigger to a visualization](#) on page 649 to learn more.

## Enabling external actions with TIBCO Cloud™ Integration

You can create external actions that start flows in TIBCO Cloud™ Integration (TCI) apps.

To be able to create and use external actions with TCI (TCI actions), you must make sure that Spotfire can access TIBCO Cloud™. Depending on your Spotfire environment, you might have to make some configurations. In this article, you learn how to do the steps necessary to enable TCI actions.

### Prerequisites

- You must have an active subscription to TCI.
- Your TCI subscription must be in one of the following TIBCO Cloud regions:
  - AWS - United States
  - AWS - Europe
  - AWS - Australia
  - Microsoft Azure - United States



If you are running Spotfire Analyst without a server, external actions with TCI are not supported.

### TIBCO Cloud™ Spotfire

If you use TIBCO Cloud™ Spotfire, you can create and use external actions with TCI without making any configurations.

Your subscriptions for TCI and TIBCO Cloud Spotfire must be in the same organization and region on TIBCO Cloud.

## On-premises environments

If you have an on-premises installation of Spotfire, an administrator must make the following preparations to enable the full use of TCI actions:

- To configure and create TCI actions. Add the TIBCO Cloud™ region where you have TIBCO Cloud™ Integration to the preference setting **Application > OAuth2Preferences > OAuth2IdentityProviders**. See *Adding TIBCO Cloud™ regions for external actions* below.
- To use TCI actions in Spotfire web clients. Register your Spotfire Server as a client in TIBCO Cloud™, and add the OAuth2 credentials to the Web Player service configuration. See [Registering your Spotfire Web Clients as a TIBCO Cloud OAuth2 client](#) in the *Spotfire® Server and Environment - Installation and Administration* user guide.

## Adding TIBCO Cloud regions for external actions

When you configure a TCI action, you must select the TIBCO Cloud region where you have your TCI subscription. The preference `OAuth2IdentityProviders` controls which TIBCO Cloud regions you can select when you configure a TCI action.

For user groups who should be able to configure TCI actions, you must make sure that the preference is configured with the TIBCO Cloud region where you have your TCI subscription.

### Procedure

1. Start Spotfire Analyst, and log in as a user with administrator privileges.
2. On the menu bar, select **Tools > Administration manager**.
3. In the Administration Manager dialog, on the **Preferences** tab, click to select the user group you want to edit preferences for.
4. On the **Preferences** tab, click **Edit**.
5. In the Edit Preferences dialog, navigate to the preference **Application > OAuth2Preferences > OAuth2IdentityProviders**.
6. To edit the `OAuth2IdentityProviders` preference, select the preference and click the edit button.
7. In the String Collection Editor dialog, enter the information for the regions you want to add.

For example, to add only the region Amazon AWS - United States, enter the following:

```
[
  {
    issuer: "https://account.cloud.tibco.com",
    displayName: "TIBCO Cloud (AWS - United States)",
  }
]
```

For information about the different regions, see the example further down on this page.

8. To save your changes, click **OK**.



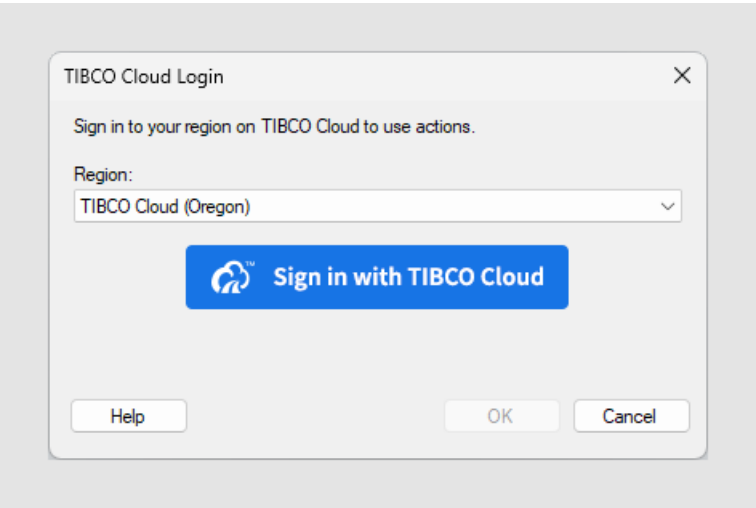
### Example with all TIBCO Cloud™ regions


Below is an example value for the `OAuth2IdentityProviders` preference with 4 different TIBCO Cloud regions added. With this configuration, you can select from all the regions in the drop-down menu when you configure a TCI action.

```
[
  {
    issuer: "https://account.cloud.tibco.com",
    displayName: "TIBCO Cloud™ (AWS - United States)"
  },
  {
    issuer: "https://eu.account.cloud.tibco.com",
    displayName: "TIBCO Cloud™ (AWS - Europe)"
  },
  {
    issuer: "https://au.account.cloud.tibco.com",
    displayName: "TIBCO Cloud™ (AWS - Australia)"
  },
  {
    issuer: "https://account.us.azure.cloud.tibco.com",
    displayName: "TIBCO Cloud™ (Azure - United States)"
  }
]
```

### TIBCO Cloud Login

When you configure or use an external action with TIBCO Cloud™ Integration, you must log in to TIBCO Cloud™. Use the controls in this dialog to start the login procedure in your web browser.




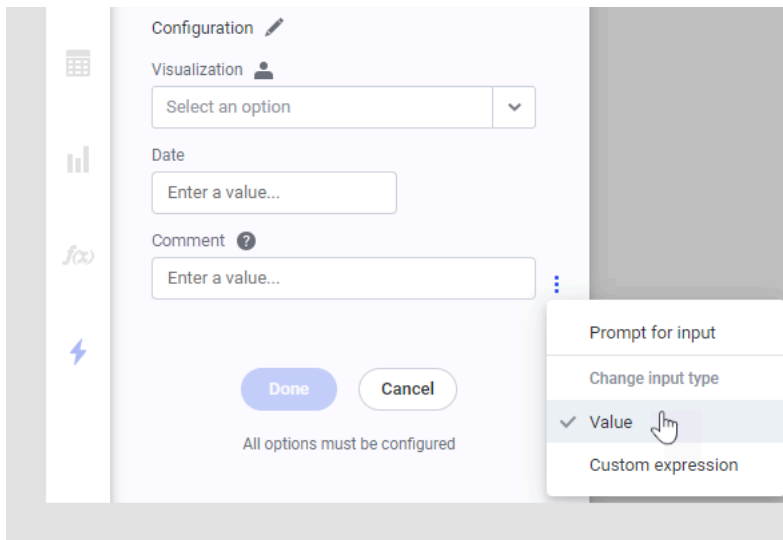
Property	Description
Region	<p>The TIBCO Cloud region to log in to. When you configure an action, you can select from a list of regions that your Spotfire administrator has configured. When you use an action, the region that you selected when you configured the action is shown.</p> <p>If you use TIBCO Cloud™ Spotfire® and you are configuring an action, only the region you are logged in to is available.</p> <div>  <p>If you are configuring an action, and you cannot see any options in the drop-down menu, your administrator must add the regions to the preference <code>OAuth2IdentityProviders</code>. See <a href="#">Enabling external actions with TIBCO Cloud™ Integration</a> on page 651 for more information.</p> </div>



Property	Description
Sign in with TIBCO Cloud	Start the TIBCO Cloud login procedure in a new tab in your web browser.

## Configuring action parameters

Which options you have when configuring an action parameter depends on the type of action and how the parameter has been defined by the developer of the action. All input types and options described below will therefore not be available for each action or parameter.

The image below shows a few parameters which might be available during configuration of an action in an action mod. Click the three dots  to the right of the parameter to see the available options.



The icon showing a person  to the right of the "Visualization" parameter means that **Prompt for input** has been selected for that parameter. The question mark  to the right of the "Comment" parameter means that a description is available for the parameter. Hover over the icon to see the description.

- **Value** - type or select an input value. How to provide input depends on the data type of the parameter.
- **Custom expression** - specify your own expression for the parameter. The Limit by option is available for custom expressions in external actions, and can be used if you want to limit the data using filterings and/or markings.
- **Document property** (available for external actions only) - select a document property to use as input. Use the search field to help locate your property.
- **Prompt for input when running action** or **Prompt for input** (depending on type of action) - select this option if you want the person running the action to provide the input. For external actions, you can change the label and add a description to each parameter. For example, it can sometimes be useful to add a description to explain what input is expected for a specific parameter. For actions in action mods, a description must be added by the mod developer. How the parameter is defined, for example, which data type is expected, will decide what kind of control will be shown in the prompt when the action is triggered:
  - Boolean (true/false) - check box.



For external actions, where you specify the labels that will be shown in the prompt during configuration, make sure that you phrase the check box label to clearly state that a selected check box means true, while a cleared check box means false.

- Predefined values (enum) - drop-down list.
- Integer - spin box where only integers are allowed.
- Real - text field where numeric values (integer or real) can be entered.
- String - text field.
- Data table - drop-down list.
- Visualization/Page - drop-down list.
- **Data view** (available for external actions only) - if a parameter has been defined with **Data view** as input type, a number of records in a tabular format will be sent when the action is run. Select which data table the parameter should retrieve data from, and in **Row for each**, select a column to split the data into categories. Then define what data to send for each of the categories by mapping each item in the parameter to a column. The **Limit by** option is also available for data view parameters, and can be used if you want to limit the data using filterings and/or markings. The image below shows an example of a parameter with data view input.

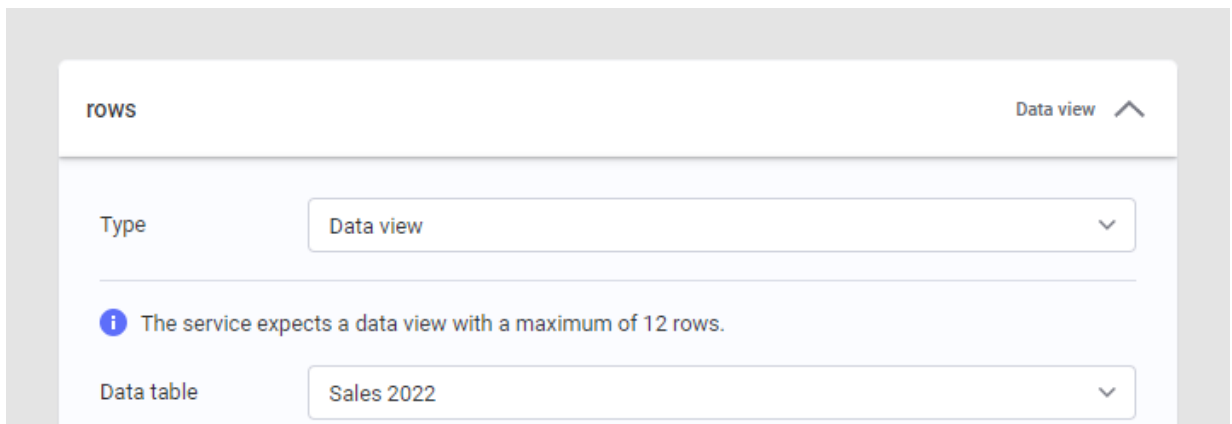
The image shows a configuration panel titled 'rows' with a 'Data view' toggle in the top right. The panel contains several input fields:

- Type:** A dropdown menu set to 'Data view'.
- Data table:** A dropdown menu set to 'Sales 2022'.
- Row for each:** A dropdown menu set to 'Customer ID' with a '+' button next to it.
- Name:** A dropdown menu set to 'First(Name)'.
- Sales:** A dropdown menu set to 'Sum(Sales)'.
- Limit by:** A section with a 'Marking' button (with an 'x' icon) and an 'Add' button (with a dropdown arrow).

- **None** - this option is available if a parameter has been defined as optional for an external action. Select **None** if you do not want to provide any input for the parameter.

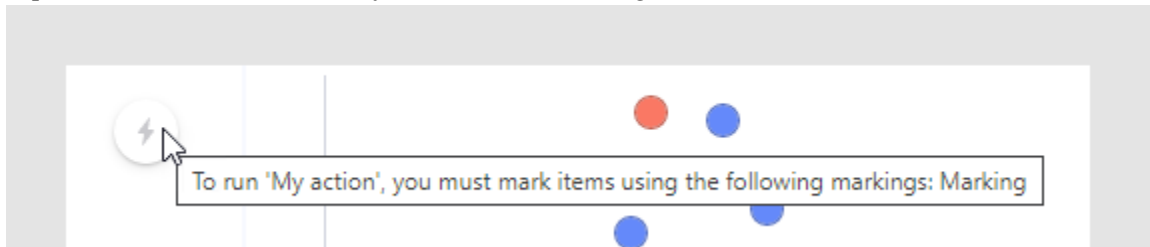
### Row limits for data view parameters

The person creating the service can specify row limits for data view parameters. This means that to run the action, the number of rows sent from that parameter must be within the limits. When row limits exist for a data view parameter, information about the limits will be shown in the Configure action flyout. The image below shows a data view parameter where only a maximum row limit has been specified, but you might also encounter services where the parameter expects a minimum number of rows, has both a minimum and a maximum number of rows defined, or expects an exact number of rows.



If you configure a data view parameter that requires a minimum or exact number of rows so that it also is limited by one or more markings, then any triggers that exist for the action will become grayed out until some items in the visualizations are marked.

A tooltip will let the person who wants to run the action know which markings must be used before it is possible to run the action, as you can see in the image below.



## Running an external action

If actions involving external systems have been added to your analysis using the installed client, you can run those actions from the web client as well.

To run an action, it must be associated with an action trigger, which can be either a floating button in the upper right visualization corner, a right-click menu option, or a button in the visualization title bar.

### Procedure

1. Locate the visualization containing the trigger of the action you want to run.
2. Use the associated trigger to initiate the action.
3. If the action has not been trusted before, you must review it in the **Review external action** dialog. If you have confidence in the signer of the action, click **Trust**.



You can select the check box **Always trust actions signed by** to trust the signer of this action. All future external actions or new versions of an action from that signer will automatically be trusted.

If the action configuration contains prompted user inputs, a dialog with one or more input fields is opened. If a question mark icon is shown above any of the input fields, a description is available in the tooltip.

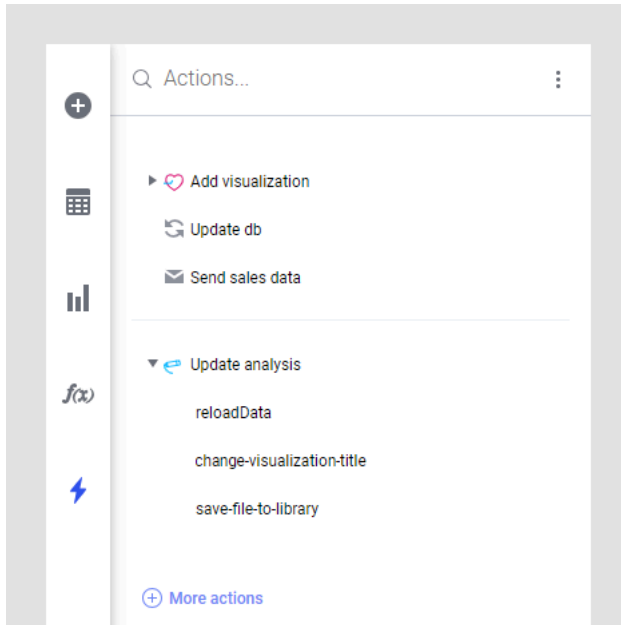
4. If a prompt is shown, enter values in the fields, and click **Next**.

The **Run external action** dialog opens. If the action is configured to send data, you will see a sample of the data that will be sent externally.

5. Review any data that will be sent with the action, and click **Approve**.  
If you are not already logged in to TIBCO Cloud™, you will be asked to log in. As soon as you are logged in, Spotfire runs the action.

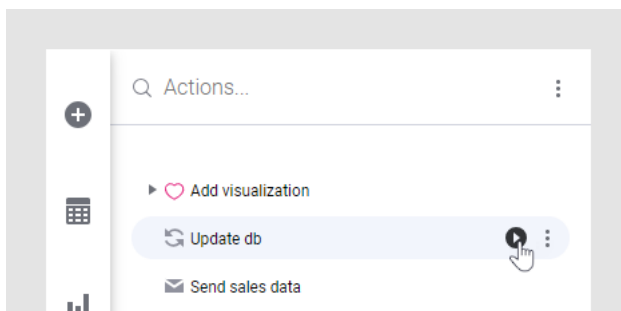
## The Actions flyout

The Actions flyout gives you an overview of all the action mods and external actions in your analysis. You can run actions ad hoc from the flyout, and pin your favorite action mods for quick access in all analyses.

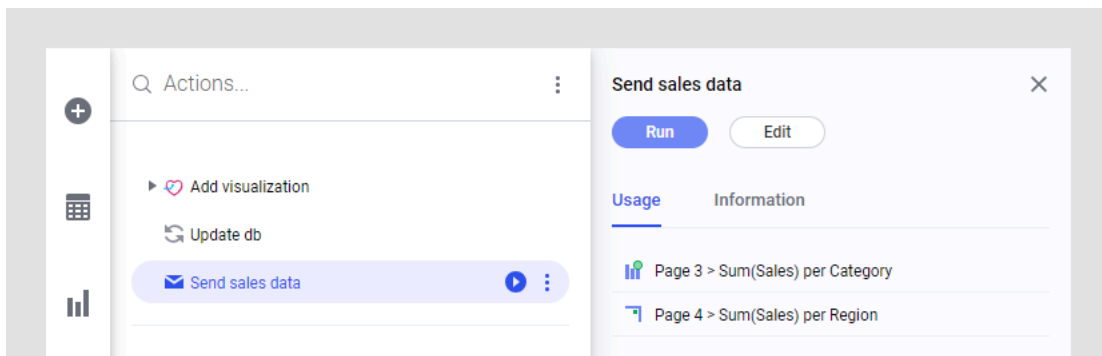


Actions that have been pinned to the flyout are shown below the line. An administrator can pin action mods to the flyout using a preference setting. See [Pinning action mods to the Actions flyout](#) on page 659 for more information.

You can run actions ad hoc from the flyout. Just locate the action you want to run, hover over it, and click the Run button to the right.



To view details of an action, you can hover over the action, open the menu to the right, and select **View details**. On the **Usage** tab, you can see where the action is used in the analysis.



On the **Information** tab, details about the selected action is shown. The signer of the action is always shown, and for action mods, information like API version and mod ID is also shown. If an action mod is saved in the library, the library path will also be shown.

### Saving an action mod in the library

You can save an action mod in the library. Once saved in the library, you can pin the action mod to the Actions flyout and run it ad hoc from the library.



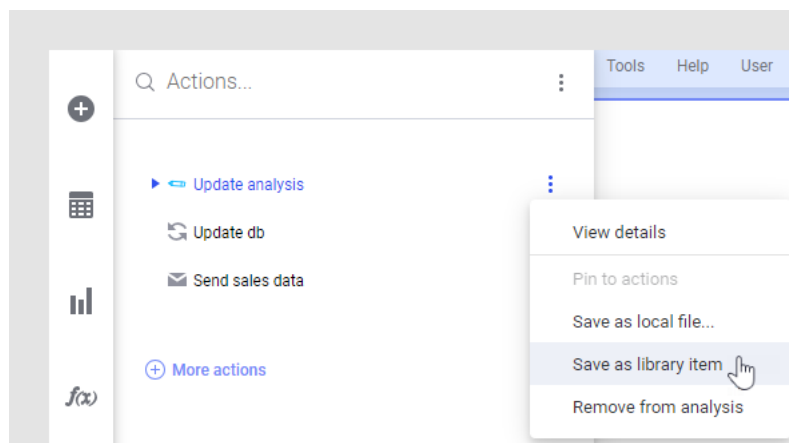
You can search the library to find action mods.

#### Prerequisites

You must have the license feature Save Action Mod to Library, the analysis must be in **Editing** mode, and the action mod you want to save must be available in the analysis but not in development.

#### Procedure

1. On the authoring bar, click **Actions** ⚡ to open the flyout.
2. Locate the action mod you want to save, and click the menu to the right ⋮.
3. Select **Save as library item**.



4. In the **Files and data** flyout, select a folder to save the mod in, type a name in the bottom part of the flyout, and click **Save**.



If you have the Develop mods license feature, you can also save the action mod to the library from the Mods development tool. Make sure the mod is disconnected, and then select **Save as > Library item**.

To find where in the library an action mod is located, open the Actions flyout, click the menu to the right ⋮ of the action mod, and select **View in library**.

## Pinning action mods to the Actions flyout

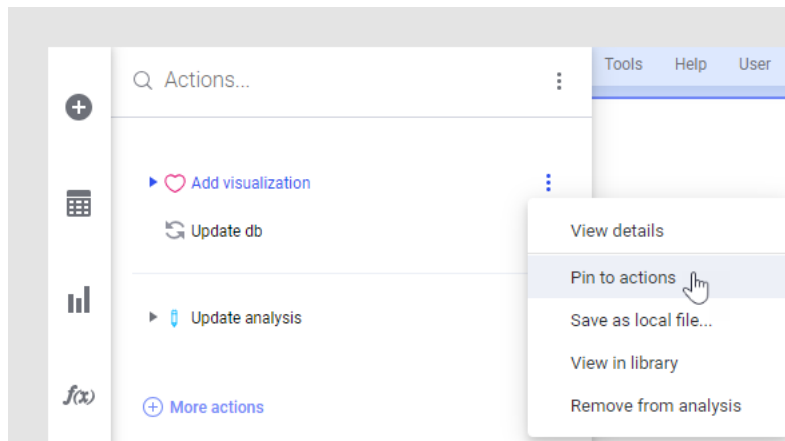
Action mods that are saved in the library can be pinned, which means they will become available for selection from the flyout also in future analyses.

### Prerequisites

The analysis must be in **Editing** mode, and the action mod must be saved in the library. See [Saving an action mod in the library](#) on page 658.

### Procedure

1. On the authoring bar, click Actions ⚡ to open the flyout.  
The upper part of the flyout, above the line, lists action mods that are not yet added to the list of pinned action mods.
2. Hover over the action mod you want to pin, click ⋮ to open the menu, and select **Pin to actions**.

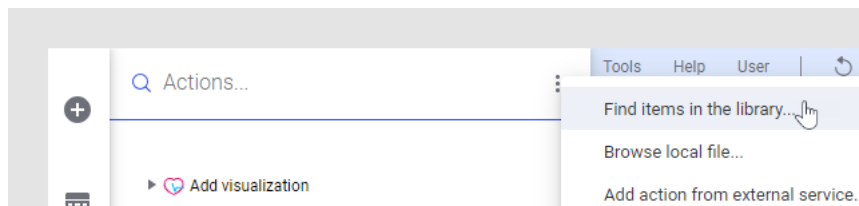


The action mod is shown below the line, in the lower part of the flyout.



If you want to remove an action mod from the Actions flyout, hover over the action mod, click the ⋮ button, and select **Unpin**.

If the action mod is not available in the flyout, you can click the menu at the top right of the flyout and select **Find items in the library**.



A search is performed in the library to show all available action mods that you can pin to the flyout (type:action).

## Running actions from the Actions flyout or the library

You can run actions ad hoc from the Actions flyout, and actions that are part of an action mod that is saved in the library can be run directly from the library.

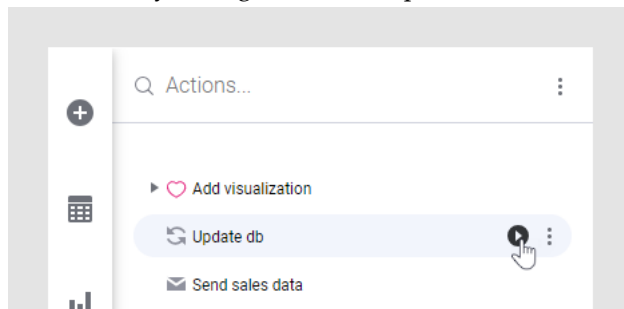
### Running an action from the Actions flyout

#### Prerequisites

Make sure data is loaded into the analysis.

#### Procedure

1. On the authoring bar, click **Actions** ⚡.
2. In the flyout, locate the action you want to run and click **Run** ▶ to the right. If the action is in an action mod, you might need to expand the item to select a specific action.



3. Depending on the action you run, in the next step you might need to provide input and click **Run** again, or approve that the action will be run. See also, [Configuring action parameters](#) on page 654 and [Running an external action](#) on page 656.



If you run an action in an action mod and want to save the configuration with the parameters you specify, you can enter a name for the configuration. The configuration will then be available on the **Test** tab of the Mods development tool as well as in the details view of the Actions flyout.

### Running an action from the library

#### Prerequisites

Make sure data is loaded into the analysis, and that the action mod is saved in the library.

#### Procedure

1. On the authoring bar, click **Files and data** +.
2. In the flyout, click **Spotfire library**, and locate the action mod.
3. Right-click the action mod and select **Run**.  
A flyout showing the actions in the selected action mod is opened.
4. Select the action you want to run, and click **Run/Next**.  
The flyout will show any parameters that must be specified for the action.



- Specify the parameters for the action and click **Run**. See [Configuring action parameters](#) to learn more about the available options.  
The action is run.



If you want to save the configuration with the parameters you specify, you can enter a name for the configuration. The configuration will then be available on the **Test** tab of the Mods development tool as well as in the details view of the Actions flyout.

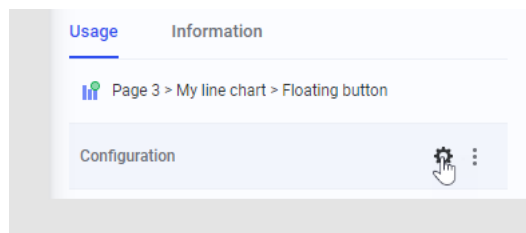
## Editing a configuration of an action in an action mod

If an action in an action mod is used anywhere in your analysis, for example on a trigger in a visualization or in an action control in a text area, a configuration has been created for that action. A configuration could also have been created as part of performing a test run of an action. You can see which usages and configurations that exist for an action in the Actions flyout. You can edit an action configuration from the Actions flyout using any client.

The analysis must be in **Editing** mode.

### Procedure

- On the authoring bar, click Actions ⚡ to open the flyout.
- Locate the action, open the menu to the right ⋮, and click **View details**.
- Open the **Usage** tab of the details view.
- Hover over the configuration you want to edit, and click the gear icon ⚙️.



- Make your changes to the parameters in the flyout, then click **Done**. For more details on the available options for the parameters, see [Configuring action parameters](#) on page 654.  
The configuration is updated and will be applied when using any trigger associated with this action configuration in the analysis.

## Configuring an external action

An external action can be reconfigured from the **Actions** flyout using any client.



The new action configuration will be applied when using any trigger associated with this action configuration in the analysis.

### Prerequisites

You must have the Configure External Actions license feature, and the analysis must be in **Editing** mode.

### Procedure

- On the authoring bar, click Actions ⚡ to open the flyout.
- Locate the action, open the menu to the right ⋮, and select **Edit action**.

3. In the Configure action flyout, make your changes and click **Next**. For information on the available options for the parameters, see [Configuring action parameters](#) on page 654.
4. Review the action configuration and click **Done**.



After you have completed the configuration, you will be the signer of the action.

## Data functions and the f(x) flyout

In addition to the possibility to add your own calculated columns or custom expressions to an analysis, a business author might also have access to data functions that have been created by a script author using Spotfire Analyst and saved in Spotfire library. These data functions can be used to add more advanced calculations and use scripts based on different programming languages in the analysis.

For more information, see the following sections:

- [What are data functions?](#) on page 662
- [Running data functions in Spotfire web clients](#) on page 664
- [Data functions in the Data canvas](#) on page 668
- [Interactive tuning of data functions](#) on page 672
- [The f\(x\) flyout](#) on page 675

For information about calculated columns and custom expressions, see [Making your own calculations](#) on page 728.

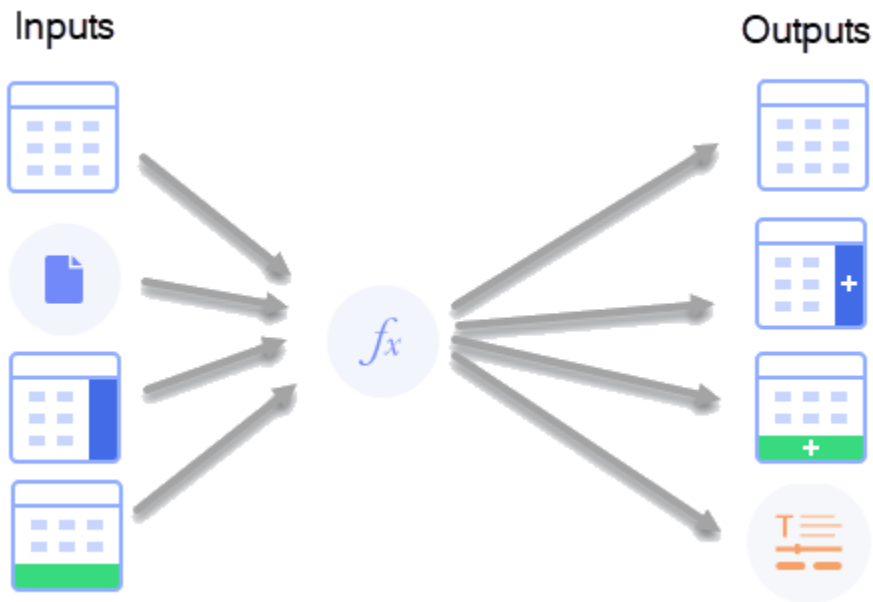
### What are data functions?

Data functions are the Spotfire way of letting advanced analysts, statisticians or mathematicians enhance Spotfire by creating scripts that can perform pretty much any type of calculation and returning the results to a Spotfire analysis. They can be created using the installed Spotfire client. If the data function is saved in the library, a Spotfire Business Author with the **Execute Data Functions** license feature (under the Spotfire Advanced Analytics licence) can use the data function when creating a new analysis. Consumer users can benefit from the results of the calculations when interacting with the finished analysis.

Due to their flexibility, data functions can be used for many different things, for example:

- Opening data.
- Transforming data (transformations can only be added using the installed client).
- Adding features to a visualization (for example, curves) by addition of a new data table, based on the first one.

In most cases, the use of a data function is a matter of mapping inputs to outputs, that is, the script requires someone to say what to base the calculations on, and where to place the result from the calculation, in the context of your current analysis.



Inputs can, for example, be a value, a column or a data table in your current analysis, but it is also possible to let the script import data from somewhere else, and have the data function as the source for the first data table in the analysis.

The output is any combination of numeric values (e.g., model coefficients, forecasts, etc.), text (e.g., summary diagnostics) or even R graphical objects. An output is also mapped to a value, a column or a data table in Spotfire. When new columns are created, they can be incorporated into an existing data table, if desired. Single value outputs can be mapped to a property and shown in a text area, if this has been configured using an installed client.

To easily find and reuse data functions from the library, they can be [pinned](#) to the [f\(x\) flyout](#).

### Data function definitions vs. data function instances

What is saved in the library is actually the data function definition. It contains the script itself and the author's specification of what types of inputs and outputs to expect or allow.

When you run a data function by mapping the definition to inputs and outputs in your analysis, you create an instance of that data function in the document. You can, in fact, have multiple different instances of the same data function in your analysis, if you run it multiple times, but the only time you would actually need that is when you run the data function with different inputs, and you want to use or keep all of the different outputs (similar to how you can create multiple calculated columns using the same function). Keeping a single instance of each data function definition in the document is most of the times the preferred option, for performance reasons.

If you just want to refresh the data function, or tweak the parameters or the script of an already existing data function instance, you can edit it from the [Data canvas](#), instead of adding it again.

### Similarity to expression functions (installed client only)

By saving a script as an expression function (only possible with scripts based on Spotfire® Enterprise Runtime for R (a/k/a TERR™)), a statistical calculation can be used in the Spotfire expression language directly, as any other function. This is a way you can enhance the expression language with your own, script-based functions. See [How to Create an Expression Function](#) in the Spotfire Analyst help for more information. In contrast to data functions, expression functions are always defined in the context of an analysis and cannot be shared between analyses (except by copying the script from one analysis and

saving it in another). Because data function definitions can be saved in the library, they are much easier for others to find and reuse.

## Running data functions in Spotfire web clients

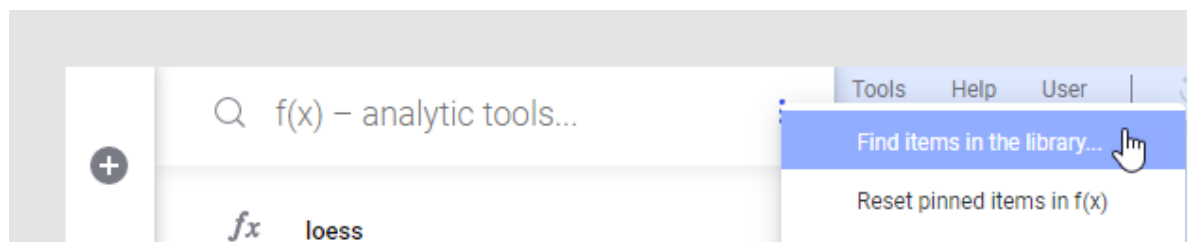
Data functions that perform different types of calculations can be created using the installed Spotfire client. If the data function is saved in the library, anyone with authoring licenses plus the **Execute Data Functions** license feature (under Spotfire Advanced Analytics) can use the data function when creating an analysis. Consumer users can benefit from the results of the calculations when interacting with the finished analysis.


### Prerequisites

Data function definitions have been created using the installed Spotfire client and saved in the library.

### Procedure

1. On the authoring bar, click **f(x)** or **Files and data**.  
If you cannot see any data functions in the **f(x)** flyout, you can select and pin data functions from the library yourself, provided that you have the required licenses. See [Pinning data functions to the f\(x\) flyout](#) on page 676.
2. Use search to find a specific data function. Note that search in the **f(x)** flyout will only search among the pinned data functions in that flyout; you must either start in the **Files and data** flyout or use the menu in the search field and select **Find items in the library** to actually search the library.



When clicking **Files and data** , you can find all data functions in the library by entering `type:datafunction` in the search field, and limit further by typing a part of the name. See [Searching the Library](#) for more details.

3. Click on the data function in the **f(x)** flyout, or double-click on the data function in the **Files and data** flyout, to run it directly.



Some data functions require that you already have some data in the analysis to be able to run. Others might work best if you have a specific type of visualization available, etc.

You can view more information about the data function before you run it by clicking on the menu next to it in the list and selecting **View details**.

4. If the data function needs some configuration, the flyout will show a list of the information you should provide. Click on each item to expand and make the selections needed.



If there are many inputs in your selected data function, and only some of them are required, you can click **Always show required inputs first** to move optional inputs to the end of the list and hide them.

It is the author of the data function that determines what the data function will do and what it needs to be able to perform its function. In some cases, you might need to select a data table, in others, you would type a value, or select one or [more columns](#). You must provide all required options to be able to continue, but you might also be able to provide optional configurations, such as threshold values or similar.

If you need more help understanding the different input or output types, see [Configuring data function parameters](#).

5. Determine whether updates to the data function results should be manual or automatic whenever input settings are changed, using the **Refresh function automatically** check box.
6. Click **OK** when you are done.
7. If the results from the data function can be added to the analysis in several different ways, the summary view is shown in the flyout and you will get the option to determine [how to add your new data](#).

Depending on the type of data function, and how the author has configured it, you might be able to add the data as a new data table, as rows or columns added to an existing data table, as calculated columns, or, for single values, as a document-, column- or data table property value. If there are multiple outputs, you can exclude some, but not all, of them.

8. Click **OK** when you are done.

## Result

The data function is executed.

Depending on the type of data function and what the results are, the analysis will be updated and ready to be opened by Consumer users once it has been saved in the library.



When a data function instance has been added to an analysis, and the outputs are columns or rows used in a visualization, you can [tweak your input values](#) directly from the visualization, and quickly try out different input values. Even more editing options are available when looking at [Data functions in the Data canvas](#) on page 668.

## Configuring data functions

When you run a data function, you must define how the input and output parameters of the data function definition should be handled within your analysis. It is necessary to specify a mapping of all required parameters to Spotfire, to use the data function in an analysis. This topic contains details about different configurations that might be available when running or editing a data function in the web client.

### Prerequisites

Data functions have been created using the installed Spotfire client and saved in the library.



All of the required inputs and all outputs (except those you explicitly exclude in the summary view) must be configured to run the data function. Optional inputs can be left without being configured.

### Procedure

1. Follow steps 1, 2 & 3 in [Running data functions in Spotfire web clients](#).
2. If the data function is designed to use input parameters to determine on what to base the calculation, you must configure your inputs.

For example, this could be a value, a column or a data table that you select from your analysis. Not all data functions require inputs, and there might also be cases where optional inputs can tweak the result from the calculation. It is the author of the data function that determines what you need to provide, by specifying the input type and what is required.



If there are many inputs in your selected data function, and only some of them are required, you can click **Always show required inputs first** to move optional inputs to the end of the list and hide them.

### Inputs:

You will not be able to select from all of the input types described below when specifying the input for a selected parameter; you only see options that are applicable to the current data function and your analysis.

- **Data table** – select a data table using the drop-down list.

(The data table selector is often a first step before selecting one or more columns, but, it can also be a separate input type. In this case, the input type only lists data tables where columns of all data types are allowed as inputs to the data function. If you cannot find the desired data table in

the list, you can instead use the Columns option to select all columns of allowed data types from a specific data table.)

- **Column** – select a single column from the specified data table using the drop-down list.
- **Columns** – [select one or more columns](#) in the Select columns dialog.
- **Search expression** – select a number of columns based on a search expression (press Enter to perform the search). This option is useful if you want to select many columns that, for example, start with the same letters.
- **Custom expression** – specify your own expression in the Edit expression dialog.
- **Value** – type an input value. The value is generally accompanied by a data type selector, where you can change the data type for the entered value.
- **None** – no input handler has been selected or no default exists. This can be used for optional input parameters. If the input parameter is required, you must specify an input to be able to continue.

The **Limit by** option allows you to limit the calculations based on column values or data tables to rows matching a specified combination of filterings and markings only. If more than one option is selected, then calculations will be performed for rows matching the intersection of the selected filtering and markings only. Do not add any limits to base calculations on all rows.



When a data function instance has been added to an analysis, and the outputs are columns or rows used in a visualization, you can [tweak your input values](#) directly from the visualization, and quickly try out different input values. You can also edit input and output parameters from the [Data functions in the Data canvas](#) on page 668 view.

### 3. Determine whether to **Refresh function automatically**.

Select this check box to update the results from the data function automatically each time the input settings are changed. If the check box is cleared, a manual refresh is needed for any updates to take effect. Refresh can be done from the Data function properties dialog in the installed client, or, a Spotfire Analyst author might add an action control to refresh from a text area.

A data function configured to load automatically will switch to manual update if cyclic dependencies are detected in the analysis.

### 4. Click OK when you are done with the inputs.

5. The configuration of outputs determines what to do with the result from the calculation. For example, you might get a new data table, new columns or rows, or, a document property value that can be used to define a line in a visualization, or similar. Which types of output are available depends on what your selected data function produces (Value, Column or Table), and what you currently have in your analysis. For example, if you do not have a data table in your analysis when running the data function, the only option you will have is to add a new data table. Once you have a data table, more options can become available.

### Outputs:

- **Add as new data table** – create a new data table.
- **Add as rows** to the specified data table. See [Adding rows to a data table](#) for more information.
- **Add as columns** to the specified data table, using a join operation. See [Adding columns to a data table](#) for more information.
- **Add as calculated columns** to the selected (final) data table.

If you have chosen to limit the input to marked or filtered rows only, the **Map result to limited rows for** option lets you specify how resulting values should be added to the data table. If this check box is cleared, the results will be added to the first rows in the specified data table, but if you have chosen to calculate results for filtered values only, you probably want to add the results

to those rows that were filtered when the calculation was performed instead. Select the input parameter to match against from the drop-down list.

- **Replace data table** – replace a previously added data table by selecting it from the drop-down list.
  - **Add as document property value** in the analysis. You can either define a new property or update an existing one. Note that if you define a new property, it will not be created until the data function has successfully finished its execution.
  - **Add as data table property value** – select a data table and create or update a data table property.
  - **Add as column property value** – select a data table, a column in that data table, and create or update a column property.
6. When you are done, click **OK**.

## Selecting multiple columns

If you need to add more than one column to a data function calculation or to an axis in a visualization, you click **Select columns** to show a dialog where you can pick all columns that should be included.

### Procedure

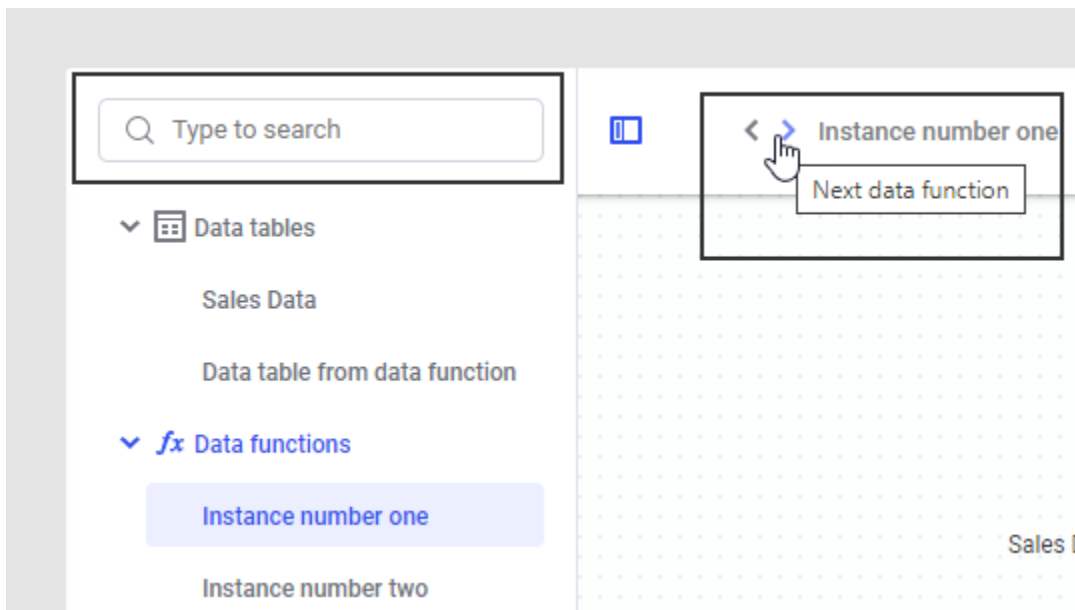
1. In the flyout when configuring a data function, first select the **Data table** to pick columns from, then click **Select columns**.  
In the context of visualizations, you can instead reach the Select columns dialog by right-clicking on an axis selector (where applicable), either directly in the visualization or from the visualization properties, and choose **Select columns** from the pop-up menu.
2. In the Select columns dialog, in the **Available columns** field, select the columns that you want to include.  
You can use the search field to limit the number of columns shown in the list. Use Ctrl + click to choose more than one column directly, or click and drag with the mouse pointer to choose many columns in a sequence.
3. Click **Add** to add the columns to the **Selected columns** list.
4. If desired, repeat steps 2 and 3.
5. In some cases, the order of the selected columns is important for the outcome. Click to choose one or more columns in the **Selected columns** list and click **Move up** or **Move down** to move them to the desired order.
6. Click **OK** (or **Close**) when you are done.

## Data functions in the Data canvas

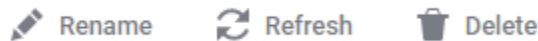
If you have added any data functions to your analysis, you can find all currently used data function instances in the Data canvas.

Use the search field in the [data canvas](#) sidebar and click on the data function instance name, or use the arrows next to the data function name, to navigate to a specific data function instance.



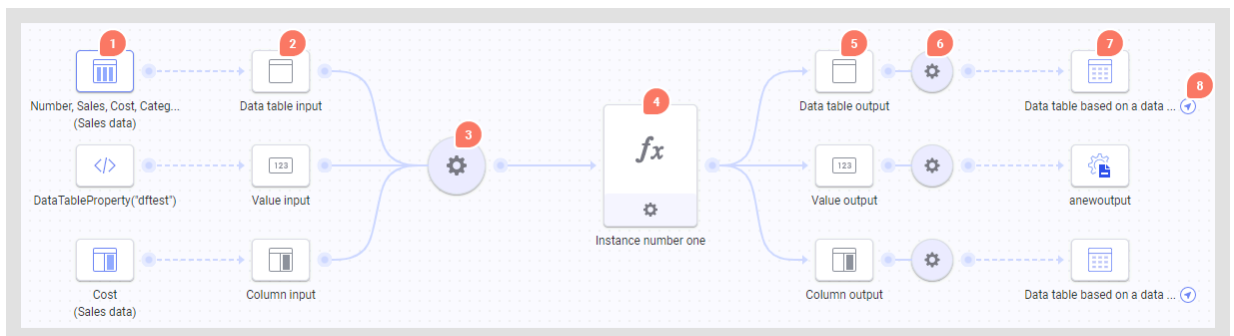


In the top toolbar, you can select to **Rename**, **Refresh** or **Delete** the data function instance.



You might have several instances based on the same data function definition in the analysis, if you have added the data function multiple times. See [What are data functions?](#) for more information about definitions vs. instances. To rename a data function definition in the library, use the Rename functionality in the **Files and data** flyout (the + on the authoring bar).

In the graphical structure, all inputs to the data function are listed to the left and all outputs to the right of the data function node ("fx" in this image). Note that the author of the data function can specify different icons for data functions when registering them, so you might see some other icon on your own data function node.



1. The leftmost column shows the currently mapped inputs and the icons represent the input type (data table, column, expression, property or manually added value).
2. The second column shows the display name of the input parameter together with the type of input as specified in the data function definition (table, column or value).
3. To edit inputs, click the round settings button.
4. The fx node represents the data function itself. Click the settings button to edit the data function script (available in the installed client only).
5. The display name and type of the outputs (table, column or value).

6. Click the round edit button to configure or reconfigure an output. Note that if you change an output used as a data source, the new output will always be applied to the end of the data source tree. It is only the modified output that is affected (if the data function contains multiple outputs).
7. The currently mapped outputs in the analysis (data table, columns or rows added to a data table, calculated column, or property).
8. The navigation arrow allows you to quickly go to the data table view of the data table where the output is used. Similarly, you can click the navigation arrow in the data table source view to go to the data function view and see all inputs and outputs for the data function instance.

You can click on a node to see details in the fields below, just as you can for data tables in the data canvas.

The **Information** tab contains information about what you are looking at right now (which type of node it is) and which configurations have been made, whereas the **Data** tab shows the actual content of the configured input or output, when applicable.


## Editing mapped data function inputs

When a data function instance has been added to an analysis, it is possible to change the configuration of the mapped inputs from the Data canvas.

### Prerequisites

Data functions have been created using the installed Spotfire client and saved in the library. At least one data function instance has been [added to the analysis](#).

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Locate the data function that you want to edit. Use the search field in the top-left selector if you have many data tables or data functions in the analysis.
3. Click the round settings button (**Edit data function inputs**) to the left of the fx node.
4. Make your changes in the Configure data function flyout and click **OK**.



You can sometimes also tweak inputs from visualizations where the data function outputs are used. See [Interactive tuning of data functions](#).


## Editing mapped data function outputs

If you have added an output result to the wrong data table, or to the wrong document property, you can reconfigure the output from the data function view in the data canvas.

### Prerequisites

Data functions have been created using the installed Spotfire client and saved in the library. At least one data function instance has been [added to the analysis](#).

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Locate the data function that you want to edit. Use the search field in the top-left selector if you have many data tables or data functions in the analysis.
3. Click the round settings button (**Configure the output**) to the right of the output parameter that you want to reconfigure.  
The Configure data function output flyout is displayed

4. Specify what to do with the output parameter, and then click **OK**.

For a description of the available input and output types, see [Configuring data functions](#) on page 666.

### Result

The function is executed. Depending on how the output parameters were set up, you may get a new or updated data table, new columns or rows, or an updated data table, column, or document property, which can be used to control a variety of settings in your analysis.

## Refreshing a data function instance in your analysis

If you have added a data function instance with manual update to your analysis, and the underlying data has been changed, you can update the calculation using the same settings as last time by clicking **Refresh** in the data canvas.

### Prerequisites

Data functions have been created using the installed Spotfire client and saved in the library. At least one data function instance has been [added to the analysis](#).




If you run a data function that creates one or more columns and then you delete one of the result columns from the analysis, the lost column remains deleted even if you refresh the data function. To recreate the lost column, edit the data function output mapping from the data canvas. This will rerun the data function and add the output result as specified.

When you add a data function in Spotfire, you can specify whether that data function should refresh automatically and run every time the Spotfire analysis is opened. You might want to avoid this option if the data input does not change, and running the data function consumes a lot of time and resources. However, if the input data changes frequently and you want the data function to reflect the changes in a visualization, automatic refresh can be helpful. You can change the **Refresh function automatically** setting at all places where you [add](#) or [edit](#) data function instances.

To refresh a data function instance manually, follow these steps.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .
2. Locate the data function that you want to refresh. Use the search field in the top-left selector if you have many data tables or data functions in the analysis.
3. Click **Refresh**.



When a data function instance has been added to an analysis, and the outputs are columns or rows used in a visualization, you can refresh the data function or change the setting to automatic refresh directly from the [tuning popover](#) in the visualization.

## Removing a data function instance from the analysis

Each time you run a data function from the **Files and data** or the **f(x)** flyout, you create an instance of that data function in the document. Keeping a single instance of each data function definition in the document is most of the times the preferred option, for performance reasons. You can remove unneeded instances from the data canvas.

### Procedure

1. On the [authoring bar](#), click **Data canvas** .

2. Locate the data function that you want to refresh. Use the search field in the top-left selector if you have many data tables or data functions in the analysis.
3. Click **Delete**.  
The data function instance is removed from the current document but any outputs that were added (columns, data tables, document properties, etc.) will remain until you remove those manually.



To remove a data function definition from the library, see [Deleting items from the library](#) on page 960.


## Renaming a data function instance in your analysis

If you use the same data function definition to add multiple instances in your analysis, for example to add multiple calculated columns based on different inputs, you might want to rename the instances to better know which instance does what.

### Prerequisites

Data functions have been created using the installed Spotfire client and saved in the library. At least one data function instance has been [added to the analysis](#).

### Procedure

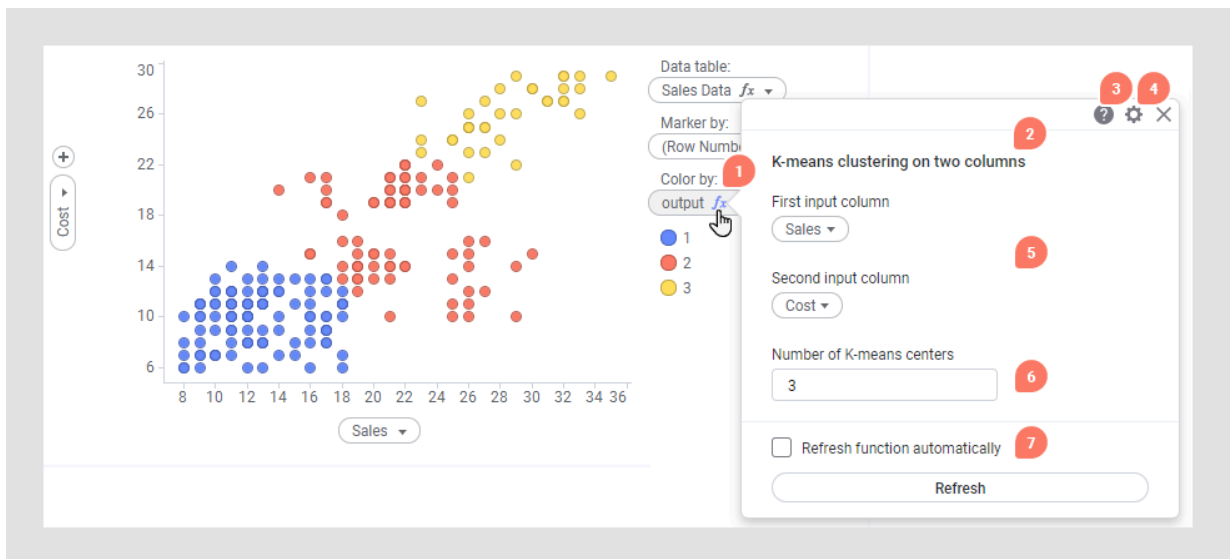
1. On the [authoring bar](#), click **Data canvas** .
2. Locate the data function that you want to refresh. Use the search field in the top-left selector if you have many data tables or data functions in the analysis.
3. Click **Rename**.
4. Type a new name and click **OK**.

## Interactive tuning of data functions

When a data function instance has been added to an analysis and the output is a calculated column, or a part of a data table (a new data table, added rows or added columns), and this output is used in a visualization, an analysis author can tune the input values directly from the visualization. This functionality is only available in Editing mode (not for consumer users).

### The tuning popover

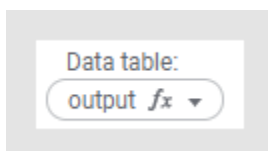
The fx icon can be seen on the column selectors or data table selectors that use the output from the data function. For example, it can be seen on the axis selector, on the color axis (in the legend), or on the data table selector.



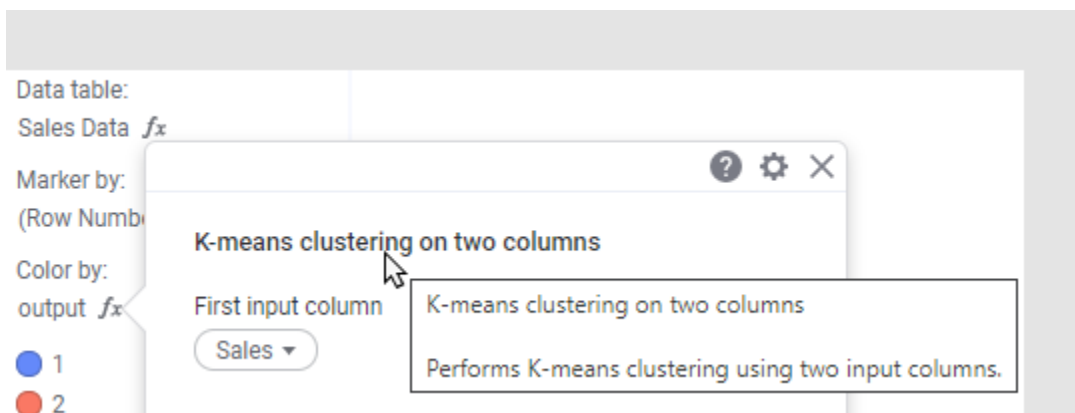
1. The access point to the tuning popover in an axis selector.
2. The title of the data function you are tuning.
3. Information about the popover and a link to this help.
4. Opens editing of the data function inputs in the configuration flyout.
5. Two examples of column inputs, showing their display names and the input controls you can modify.
6. An example of a value control and the display name of the input.
7. The option to either refresh automatically or to do a manual refresh is at the bottom of the popover.



Only data function outputs added as calculated columns will be seen directly on the column selectors of the visualization axes. To change the inputs for data added as columns, rows or data tables, you use the tuning popover on the data table selector in the legend.



When the tuning popover is open, you can hover with the mouse pointer over the name of the data function, or over the input value name, to see a description (if one has been added by the data function author):



You can also change whether the data function should refresh automatically after changes, or if you want to refresh manually. If you have not enabled automatic refresh, you can click Refresh in the popover to see the updated result directly in your visualizations.



The results from a data function might be used in several different visualizations and/or data tables, and editing its parameters will affect all places where the data function is used.

The tuning popover can be dragged to other parts of the analysis to allow you to see the entire visualization while testing different input values.

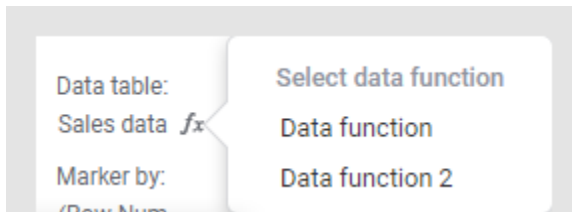
Close the popover by clicking on the x or clicking outside the popover.

If you need to change more things than those available in the popover, click the Settings icon to open the inputs for editing in the configure data function inputs flyout. You can also edit all parameters from the [data function view in the Data canvas](#).

You can always right-click on a control and select **Custom expression** to change the input value using your own custom expression. This way, you can actually switch a value input to a column input, if you know what you are doing, or change the data table to pick your columns from.

If you add more than two columns, they will be grouped together. You can click the plus sign to add more columns. For selectors that accept multiple column inputs, you can right-click and choose the **Select columns** dialog instead, to add many columns simultaneously.

If a data table or a column selector contains outputs from several different data functions, the tuning popover will first allow you to choose which data function to tweak, and then show the input controls for that data function.



## Restrictions & considerations

- It is only previously added input values of the data function instances in the analysis that you can tweak from the popover. You cannot add previously ignored optional inputs, or change any 'Limit by'-settings, or outputs here. Neither can you remove an optional parameter from this access point, if a value has been added to it already. To do such changes, you must edit the input parameters in the configuration flyout (available from the Settings button in the popover or from the data canvas). Output editing is only available from the Data canvas.
- Note that the controls used in the tuning popover might change if you interact with them, or if you close and reopen the popover. For example, if you change an input from a regular column input to a custom expression, the control will be changed from a column input handler to the custom expression handler. Likewise, if an entered custom expression is simple enough it will be reinterpreted as a column input after closing and reopening the popover.
- It is only the lowest level of each control that is shown in the tuning popover, so if you must change the data table from which to select a column, you must either go to full editing or use the custom expression editor.
- Only columns of the allowed data type, as specified by the data function author, will be shown in the column selectors.
- The fx access point is available in all places where the column selector uses web technology. That means that it is available on axes and in the legend in both web and installed clients, but it is only available from within the properties controls when using the web client (not within Visualization Properties in the installed client).

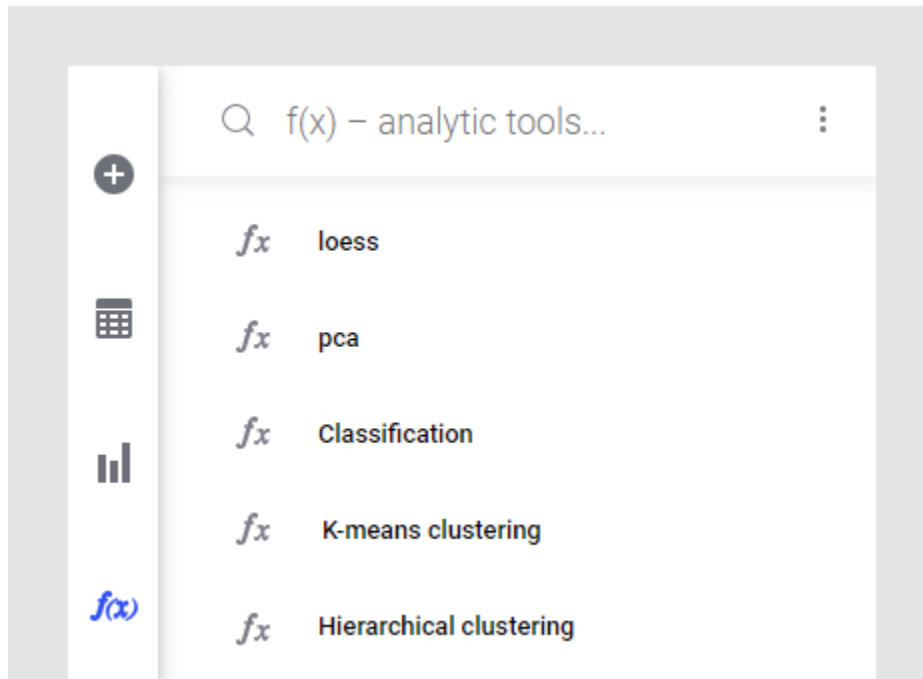
- You cannot cancel a change you make in the tuning popover, but you can always Undo the operation.

See [Configuring data functions](#) on page 666 for a description of the different types of input controls that data functions can have.

## The $f(x)$ flyout

The  $f(x)$  flyout is a handy way for Spotfire authors to easily reach their favorite analytic tools by providing the possibility to pin data functions from the library to the flyout. It is available to analyst and to business author users with the  **$f(x)$  Flyout** license feature (under Spotfire Advanced Analytics). Click on a pinned data function in the list to run it in your analysis (when applicable; for example, some data functions cannot run unless you have data in the analysis).

In the installed client, the  $f(x)$  flyout also contains the Spotfire tools for Data relationships, K-means clustering, Line similarity, Hierarchical clustering, Regression modeling and Classification modeling.



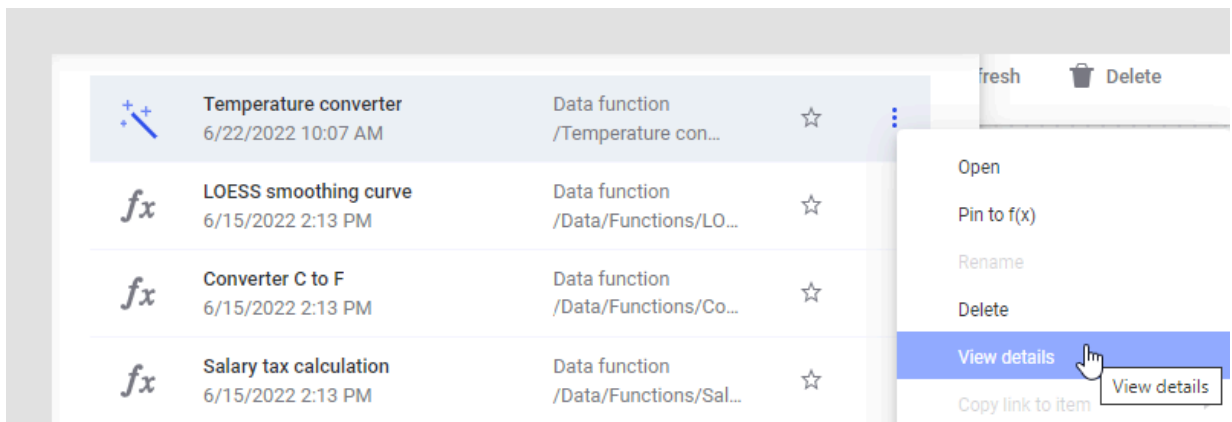
By pinning your most used data function definitions to the flyout you will always be able to easily apply the data function to new analyses.

An administrator can also pin data functions to the flyout using a preference setting. See [Pinning data functions to the  \$f\(x\)\$  flyout](#) for more information.



When a data function instance has been added to an analysis, and the outputs are columns or rows used in a visualization, you can [tweak your input values](#) directly from the visualization, and quickly try out different input values. You can also edit both [inputs](#) and [outputs](#) from the data canvas.

When hovering with the mouse pointer over a data function in the  $f(x)$  flyout, or when browsing the library using the **Files and data** flyout, you will see a menu next to the data function where you can select **View details** to see more information.



## Pinning data functions to the f(x) flyout

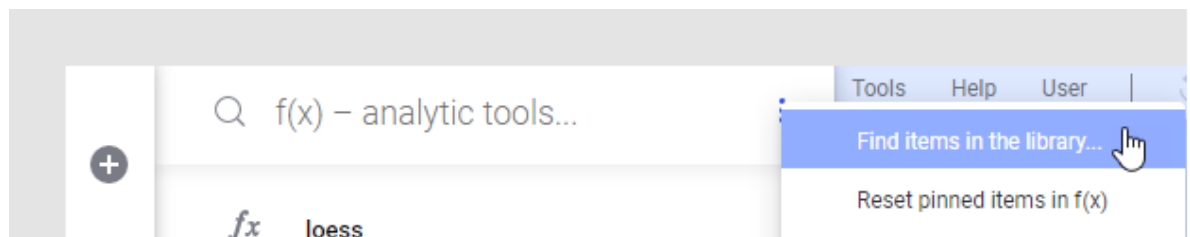
The f(x) flyout is used to provide easy access to commonly used data functions. As an analyst or a business author user with the **Pin Data Functions** and the **f(x) Flyout** license feature (under Spotfire Advanced Analytics), you can pin your own selection of data functions from the library to the flyout. An administrator can predefine selected data functions using a preference.

### Prerequisites


You are an analyst or business author user with the **Pin Data Functions** license feature. If an analysis is open, it must be in **Editing** mode.

### Procedure

1. In the **f(x)** flyout, click the menu at the top right of the flyout and select **Find items in the library**.



A search is performed against the library to show all available data functions that you can pin to the flyout (type::datafunction).

2. If desired, you can add more information to the search expression, for example, the first letter in the name of the data function you are looking for.  
See [Searching the library](#) for more information.
3. When you have located the data function of interest, click the menu next to the data function and select **Pin to f(x)**.  
The data function is pinned to the flyout. You can also pin and unpin data functions when searching the library for data functions in the **Files and data** flyout , either using the menu, in the details, or, using right-click.

See also [Unpinning data functions from the f\(x\) flyout](#) on page 677.



## Pinning data functions to the f(x) flyout using administrator preferences

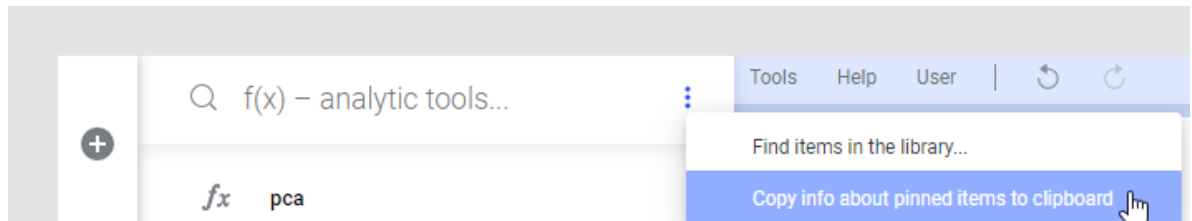
As an administrator with access to the Administration Manager, you can pin data functions and help users in different groups finding their tools.

### Prerequisites

You are an administrator with access to the installed Spotfire client.

### Procedure

1. As an administrator, follow the steps above to pin or unpin data functions from your current f(x) flyout, so that the selection of data functions in your flyout is as you want it to be for the group.
2. When you are satisfied, in the f(x) flyout, click the menu at the top right of the flyout and select **Copy info about pinned items to clipboard**.



This action copies the current layout of the f(x) flyout to the clipboard in a format that is directly suited to be pasted into the preference as described in the next steps.

3. In the installed client, click **Tools > Administration manager**, and go to the **Preferences** tab.
4. Select the group for which to change the preferences and, in the lower right part of the Administration Manager, click **Edit**.
5. Expand **DataFunctions**, click **DataFunctionsPreferences** and, next to **PinnedFunctionDefinitions**, paste the previously copied string.  
See the [Administration Manager](#) help for more information about preferences.
6. Click **OK** and **Close** the Administration Manager.

## Unpinning data functions from the f(x) flyout

The f(x) flyout is used to provide easy access to commonly used data functions. As an analyst or a business author user with the **Pin Data Functions** and the **f(x) Flyout** license feature (under Spotfire Advanced Analytics), you can pin or unpin your own selection of data functions from the library to the flyout. An administrator can predefine selected data functions using a preference.



You can quickly remove all items you have pinned and only show items pinned by an administrator. In the f(x) flyout, click the menu next to the search field and select **Reset pinned items in f(x)**.

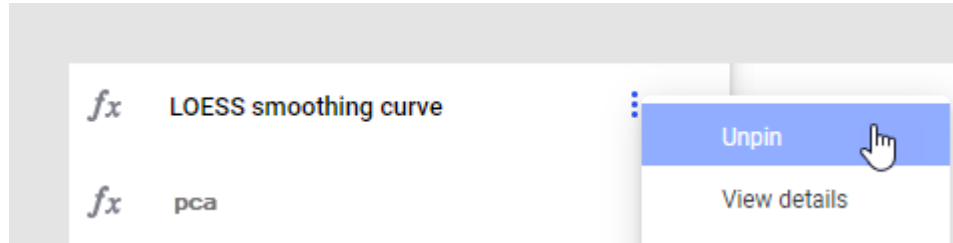
### Prerequisites


You are an analyst or a business author user with the **Pin Data Functions** license feature. If an analysis is open, it must be in **Editing** mode.

### Procedure

1. In the f(x) flyout, locate the data function to unpin.  
See [Pinning data functions to the f\(x\) flyout](#) on page 676 for information about pinning, and [The f\(x\) flyout](#) on page 675 for general information about the flyout.

2. Click the menu next to the data function and select **Unpin**.



The data function is removed from the flyout. You can also pin and unpin data functions when searching the library for data functions in the **Files and data** flyout , either using the menu, in the details, or, using right-click.

## Usage of scripts and data functions

Various scripts like IronPython scripts, JavaScripts, or scripts written in custom query languages to be executed in external databases can be part of an analysis. Addition of data functions is another way to enable scripting in Spotfire.



Scripts in action mods use different security mechanisms than IronPython scripts, JavaScripts and data functions. See [Developing mods](#) and the [Spotfire Mods website](#) on GitHub for more information about mods.

See [What are data functions?](#) on page 662 and [Introduction for Data Function Authors](#) in the *Spotfire Analyst User Guide* for more information about data functions.

If, or under which circumstances, scripts and data functions are allowed to be used depend on whether the user is working in an on-premises or cloud environment, and in which user role or product, Spotfire Consumer (web client user), Spotfire Business Author (web client author), or Spotfire Analyst (installed client author).

Also, users can be of different types depending on their authorities controlled by which groups they belong to, and the licenses specified for those groups. Licenses and groups settings are configured in the installed client under **Tools > Administration Manager** or using the web administration pages on the Spotfire Server.

The tables below list what is allowed to do for different combinations of user types and Spotfire products.

### On-premises and Spotfire Cloud Enterprise users

Single-tenant cloud environments can be configured with licenses in the same way as on-premises solutions with a Spotfire Server. Users running the installed Spotfire client without a server have the same functionality as users running Spotfire Analyst with a server and with script author license, but there is always local execution and no library or script author functionality available.

Untrusted Spotfire® Enterprise Runtime for R (a/k/a TERR™) scripts might execute in T-REX\* mode for all user types.

The TIBCO Spotfire Statistics Services server (SSS) is currently the only option to run data functions based on MATLAB or SAS.



The TIBCO Spotfire Statistics Services server is discontinued and will no longer be updated. You can continue to use it but no bug fixes or security patches will be provided, and it will eventually be removed. See the [Migration Guide](#) for more information.

Type of user	Available data function engines	Create script	Add script to analysis	Save trusted script in library	Use results in analysis
Analyst without Author Script license	Local: IronPython JavaScript TERR	No	Yes, if the script is saved as trusted to the library by an authorized** script author.***	No	Yes
Analyst with Author Script license	Python Remote: TERR (TERR Service or SSS)\$ Python (Python Service)\$	Yes	Yes, if the script is trusted by you, or saved as trusted to the library by an authorized** script author.	No	Yes, if a local engine is available or if SSS is used.\$
Analyst with Author Script license + member of the Script Author group	R (Spotfire Service for R or SSS)\$ On-prem remote only: MATLAB (SSS)\$ SAS (SSS)\$	Yes	Yes, if the script is trusted by you, or saved as trusted to the library by an authorized** script author.	Yes	Yes
Business Author with Execute data functions license	Remote: IronPython JavaScript TERR (TERR Service or SSS)\$ Python (Python Service)\$	No	Yes You can only run scripts that are saved as trusted by an authorized script author.	No	Yes You can only run scripts that are saved as trusted by an authorized script author.
Consumer	R (Spotfire Service for R or SSS)\$ On-prem remote only: MATLAB (SSS)\$ SAS (SSS)\$	No	No	No	Yes You can only run scripts that are saved as trusted by an authorized script author.

\* TERR Restricted Execution Mode (also known as T-REX mode) allows a limited subset of low-risk functions to execute unrestricted.

\*\* Script authors are authorized, if they are members of the Script Author group.

\*\*\* If you previously had a license, the scripts that you created by that time can still be run.

\$ Server-side execution requires additional products.

\$\$ The Spotfire Service for R does not provide a local engine so any R scripts executed with the R service always run on the server. Because of this, users who are not members of the Script Author group cannot run R scripts-even their own-unless the script has been trusted by a script author. However, a Spotfire Server administrator might bypass this security measure. See the *Spotfire® Service for R Installation and Administration guide* for more information.

## TIBCO Cloud Spotfire users

The general multi-tenant Spotfire cloud solution allows self-service script usage for Analyst users only. On the web, only TERR data functions are available. They always execute in the containerized TERR Service.

Type of user	Available data function engines	Create script	Add script to analysis	Save trusted script in library	Use results in analysis
Analyst	Local: IronPython JavaScript TERR Python	Yes	Yes, if the script is trusted by you.  All TERR and Python data function scripts run locally (no server-side execution permitted).  Untrusted TERR scripts can execute locally in T-REX mode.*	No	Yes
Business Author	Remote: TERR Python	No	For security reasons, scripts cannot be used, except for TERR data functions.	No	Yes
Consumer		No	No	No	Yes

\* TERR Restricted Execution Mode (also known as T-REX mode) allows a limited subset of low-risk functions to execute unrestricted.

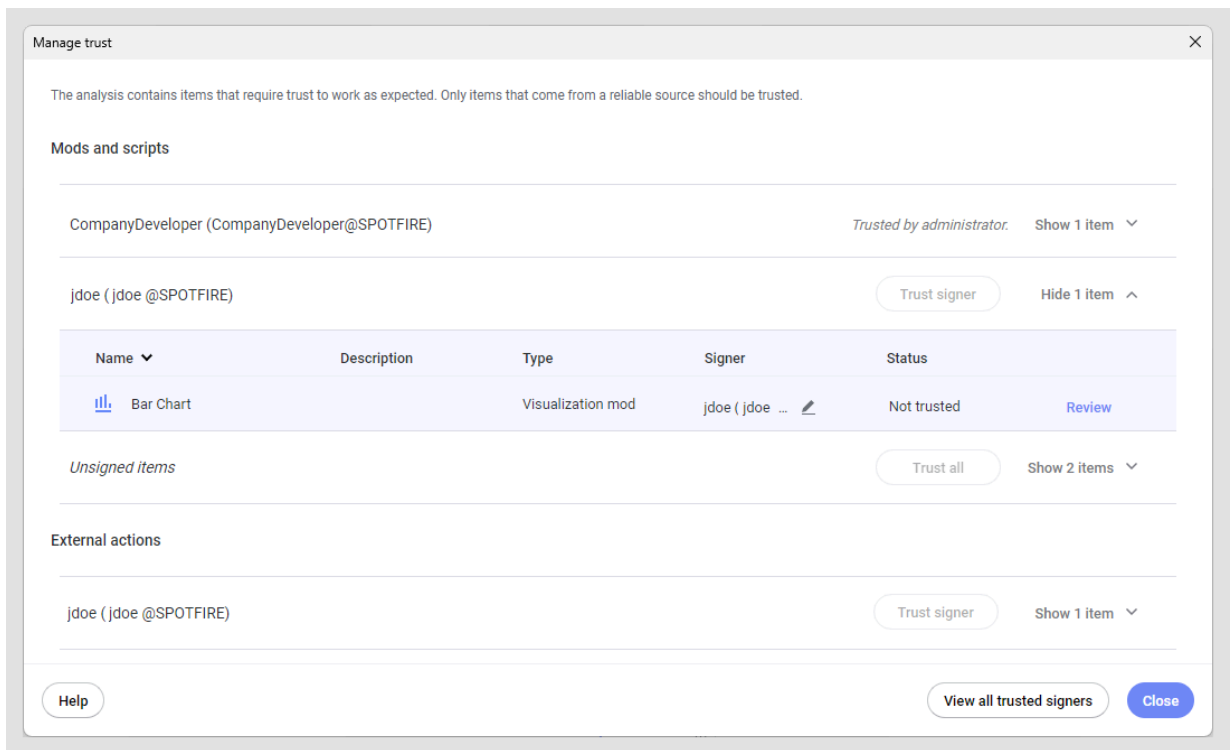
## Manage trust

Many Spotfire users want to extend the Spotfire environment in different ways. When allowing custom code to run within an analysis, or when enabling direct interaction with other systems, it is important to consider security. Any custom item created by a malevolent person could potentially perform unexpected or undesired actions. Therefore, Spotfire uses different trust mechanisms to help you to keep your system safe.

In on-premises environments, administrators can control which users should be allowed to produce custom content, verified by licenses and group belonging, and trust can be predefined. If no predefined trust is available you will get a warning when custom items are used.



All custom items in an analysis that require some sort of trust can be inspected under **File > Manage trust**. You should only trust items if you are certain that they come from a reliable source.



## Spotfire visualization mods, action mods and external actions

Spotfire mods can be created and also uploaded to the Spotfire library by a user with sufficient privileges, and, similarly, an [external action](#) that can potentially send data or interact with an external system can be configured by users with sufficient privileges. An on-premises Spotfire administrator has access to many tools to ensure that only trusted developers or configurators are allowed to add and execute code, or to configure external actions. See [Trusting custom content in the Spotfire environment](#) in the *Spotfire Server and Environment - Installation and Administration* guide for more information.

As an end user, you might have different options to trust items added by others, depending on your role in the environment.

### Signed items

Anyone who creates or adds a mod or an external action to the Spotfire environment can sign it. The signing informs other people about the origin of the item and makes it possible to make informed decisions regarding whether the item can be trusted or not. Signed items make it possible to verify the authenticity, integrity and publisher of the code or action.

Signing of mods can be done, either through certificates created by a certificate authority (CA), or automatically, using the Spotfire account of the person who loads a mod project to an analysis file. When you are offline, you can only sign mods using a certificate from a CA, not using a Spotfire account. See [Spotfire Developer Documentation > Spotfire Package Builder](#) for more information about signing mods using a certificate.

External actions are always signed with your Spotfire account.



By default, only valid signatures can be trusted. In special cases, an administrator in an on-premises system can relax this limitation by changing a preference in Administration Manager (**Application > Trust > Require valid signature to allow trust**). Users running Spotfire Analyst without a server are always allowed to trust invalid signatures and should therefore be extra careful.

### Trusting mods, actions, or signers

When a mod or external action is signed, it is easier to decide if you dare to trust it; that is, you base your trust in the company or person who has signed the item. It is possible to either trust all mods or external actions added by a certain person, that is, to trust the signer, or to trust specific items only.

If you trust a specific mod version, the mod will be seen as trusted in all analyses where it exists, however, re-trusting will be required if any changes are made to the mod at a later stage. An external action is always configured within the context of an analysis and will need to be reconfigured, and therefore signed again, if the analysis changes so that the configuration could lead to a data leak. If you decide to trust the signer, instead of a specific item, then all future items (or new versions of a mod) from that signer will automatically be trusted.

The trusting can be performed on an individual level by end-users who have permission to trust items, but an administrator can also define trust for a group of users in the Spotfire environment. To avoid unnecessary prompts regarding trust, the latter is often preferred, when possible.

### Untrusted mods

Regardless if the mod is signed or not, attempts to add a mod that is not trusted to an analysis will lead to the question of whether or not it should be trusted (or, if you do not have permission to trust, it cannot be added). Mods should only be trusted if you are certain that they come from a reliable source.

### Untrusted actions

If an action is added to an analysis, and the mod developer or configurator has not been added as a trusted signer by the administrator, clicking on the trigger for the action in a visualization (for example, a floating button or the pop-up menu) will ask you whether you trust the action and, in the case of external actions, show you which data will be affected by the action.

### Revoke trust

If you have the permission to trust signers and items you can also revoke trust you have added to an analysis using the **File > Manage trust** dialog. The **View all trusted signers** button in the dialog takes you to the **My account** page on the server, where you can get an overview of all trusted signers and items, and revoke trust that has not been assigned by the administrator. Note that an administrator can withdraw trust for something that you have trusted, or invalidate a user's signature at any time.

### Invalidate signature

If your user account has been used to sign items that you do not wish to stand behind, you can invalidate all your signatures from a specific time and up until now. This is done from the **My account** page (if any signatures are available).

An administrator can also revoke the certificate for a signer to make a signature invalid, or block a signer or a specific mod, to prevent you from adding it.

### Trust for script written in IronPython, JavaScript or within data functions

IronPython and JavaScript scripts and data functions do not support signing, so those items are always shown under Unsigned items in the Manage trust dialog. Instead, Spotfire uses a trust mechanism, where users called Script Authors, verified by licenses and group membership, are the only ones that can make a script trusted for anyone in the organization.

In the web clients, it is not possible to assign trust to such scripts. If you encounter an analysis with untrusted scripts you must either open the analysis in the installed client and trust the scripts before saving the file, or contact a script author to do it for you.

### Script authors

As a script author, you have the responsibility to provide other users in your company with secure and working scripts. When you have developed a data function or an analysis including a script you must make sure the script or data function is trusted before you save it to the library, to ensure that it can be used by others. You can review all scripts and data functions in an analysis and trust them one by one, or approve all your scripts in an analysis by selecting **File > Manage trust** and clicking **Trust all**.

If you need to edit the scripts before trusting them, go to **File > Document properties > Scripts or Data > Data function properties** where you can see an overview of all scripts and data functions, respectively.

By saving scripts in an action mod instead of just adding them with IronPython in the analysis, you can make trusting and sharing of scripts easier for both administrators and other users.

### Spotfire Analyst authors

If you are an analysis author, there can be occasions when you want to run a script or data function that has not been approved and trusted by a script author. For example, you might receive a local file from a colleague for testing purposes.

If you open an analysis with untrusted scripts, you have the option to review and trust the scripts or data functions yourself, using the Manage trust dialog. If you need to edit the scripts before trusting them, in the installed client, go to **File > Document properties > Scripts or Data > Data function properties** where you can see an overview of all scripts and data functions, respectively. You can also edit the script for a data function instance from the [data canvas](#).

If you are not the primary author of the script and you are not qualified to understand whether the script is safe, you should only trust it if you are certain that it comes from a reliable source.



In some situations, you might need to reload data after a script or data function has been trusted to get the correct data.

### Web client users

In the web clients, it is not possible to assign trust to a script, only to action mods. If you encounter an analysis with untrusted scripts you must either open the analysis in the installed client and trust the scripts before saving the file, or contact a script author to do it for you.

### Data functions written in R

Spotfire has its own implementation of the R language, Spotfire® Enterprise Runtime for R (a/k/a TERR™), which is included in Spotfire applications. TERR comes with a restricted mode which is built to provide a secure environment when working with data functions. If the data function is trusted, then it can be executed without any restrictions. If a TERR-based data function is not trusted, Spotfire will make an attempt to run the data function in the restricted mode. If the script uses statements that are not available in the restricted mode, then the data function will be prevented from running until it has been trusted.



For inline TERR scripts (that is, when using TERR directly in custom expressions, only the restricted execution mode is available).



If you have the Author Scripts license feature, but you are not a member of the Script Author group, you cannot run R scripts that you have created yourself (using the installed client) when the Spotfire® Service for R is used. See also [Usage of scripts and data functions](#) on page 678.

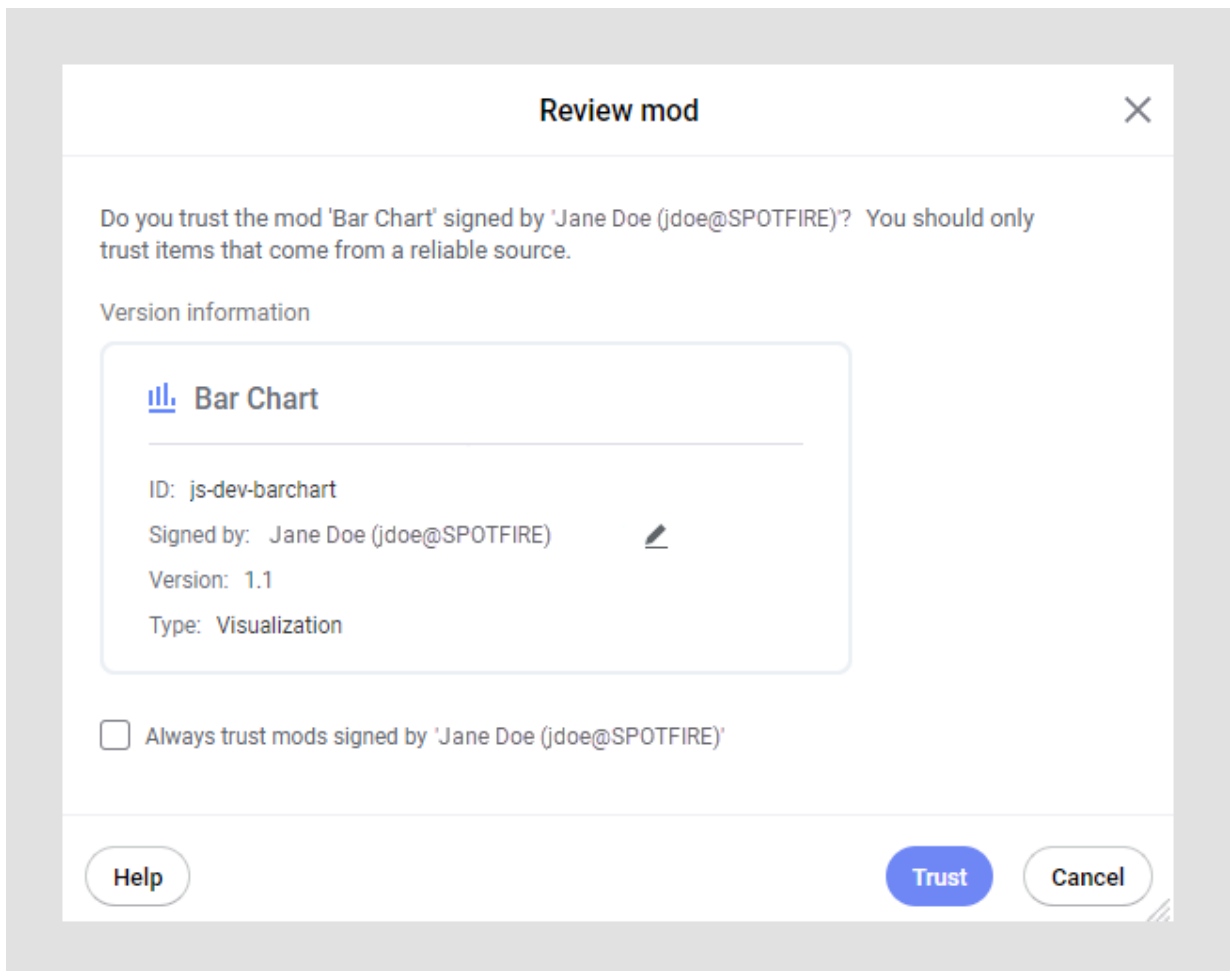
## Review mod

A visualization mod is a visualization that has been developed by someone else than Spotfire, and an action mod contains one or more scripts that you can trigger from a visualization or on a property change. You might see the question of whether you trust the mod when trying to add a new visualization or action mod to your analysis.

For security reasons, the Review mod dialog is shown if you try to add a mod that has not been trusted before. If you have the appropriate permissions, you can make the decision to trust the mod yourself.



It is important to only trust items that you are certain comes from a reliable source. Read more about trust and security in Manage trust.



If you decide to trust the signer, instead of the specific mod, then all future mods or new versions of a mod from that signer will automatically be trusted.

You can change your trust settings at any time. For more information, see [Manage trust](#) on page 680.

## Review external action

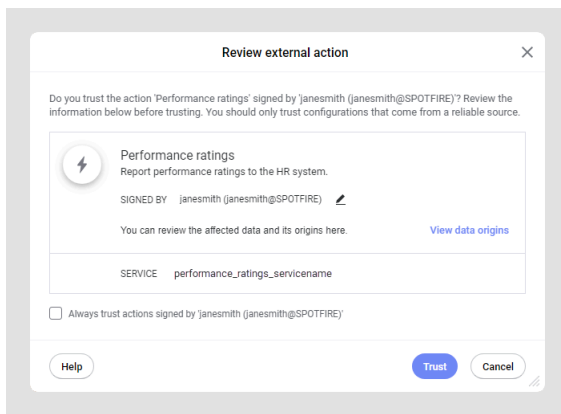
An external action is an action triggered from your Spotfire analysis which can send data to, or interact with, external systems. When you attempt to run an action, you might see the question of whether or not you trust the external action.

For more information, see [Actions](#). For security reasons, the **Review external action** dialog is shown if you try to run an external action that has not been trusted before. If you have the appropriate permissions, you can make the decision to trust the action yourself.



It is important to only trust items that you are certain come from a reliable source. Read more about trust and security in [Manage Trust](#).





If you decide to trust the signer, instead of the specific action, then all future external actions or new versions of an action from that signer will automatically be trusted.

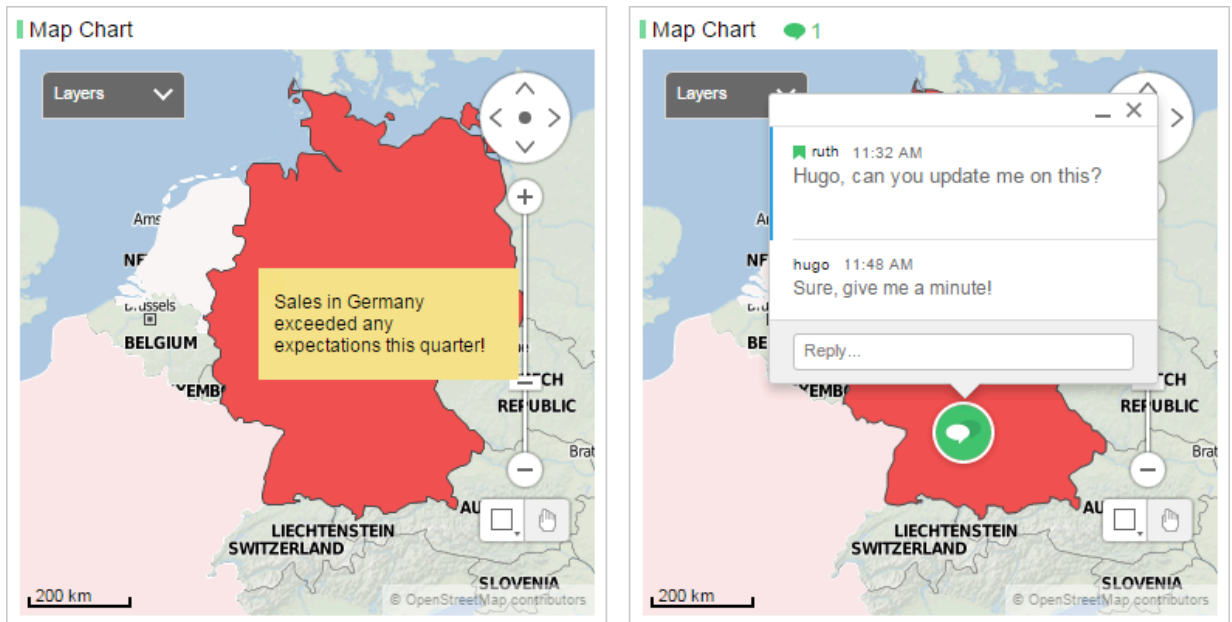
You can change your trust settings at any time. See [Manage Trust](#) for more information.

See also [Running an external action](#) on page 656.

# Annotations and conversations

Within an analysis, you can communicate in different ways. You can add comments on visualizations using annotations, or add comments to conversations to communicate. When to use annotations, and when to use conversations is described in this topic.

The visualizations below show the difference between using [annotations](#) and using [conversations](#).



## Using annotations


Annotations are text messages that are added on top of visualizations. They are added for information or presentation purposes to, for example, explain what is visualized, point to interesting findings, or give instructions. In other words, the use of annotations is a one-way communication.

You can add annotations by selecting **Visualizations > New annotation** on the menu bar.

It is possible to show more than one annotation on top of a single visualization. Moreover, by adding subsequent annotations, you can guide others through an analysis.

## Using conversations

Adding conversations on top of visualizations is a two-way communication. In a conversation, you can collaborate on an analysis with others by exchanging comments. An on-going discussion can be held on a certain matter, and everybody with access to the file can contribute.

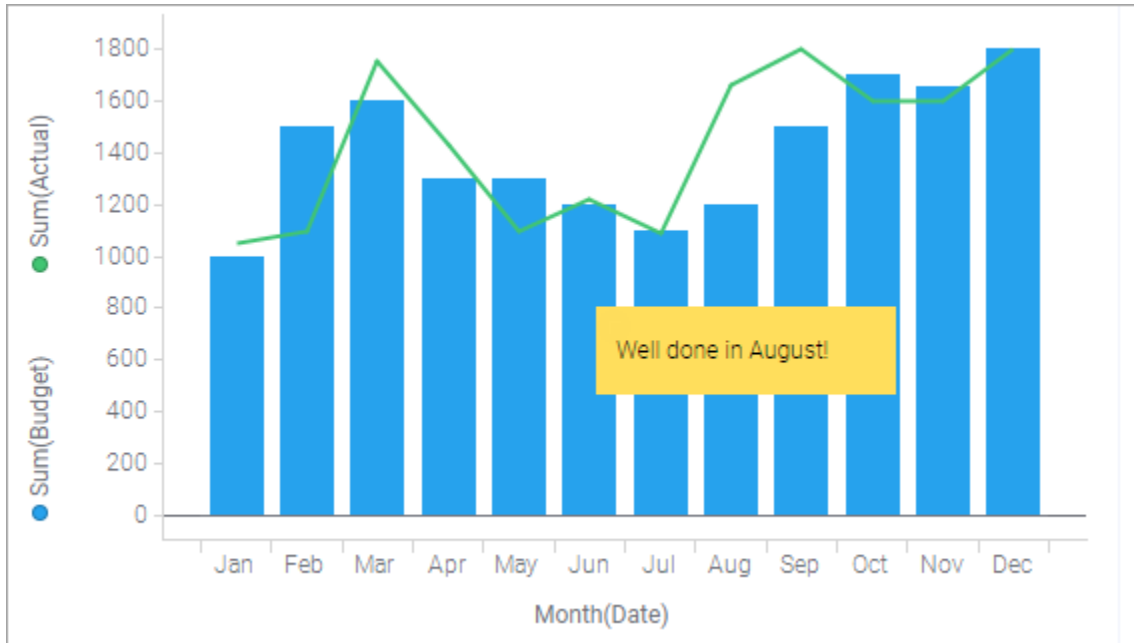
You can access conversations by clicking  on the menu bar.

Several conversation can be associated with a visualization, but only one conversation can be visible at a time.

## Adding annotations

Annotations are text boxes that can be added on top of a visualization. You can annotate visualizations for various presentation purposes, like to explain what is shown or to inform of interesting findings. They can also be used for adding instructions or questions.

An example is shown below.



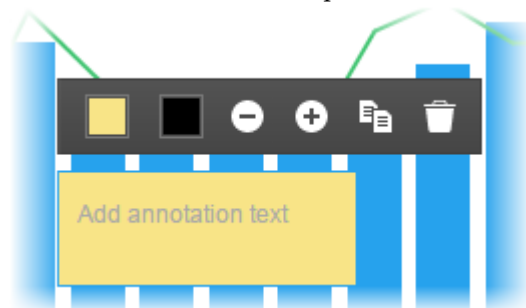
An annotation is a part of the visualization. This means that if you move an entire visualization to another position in the analysis, or duplicate it, any added annotations are included. If you export a visualization, you can specify whether to include the annotations.

The position of an annotation can be changed. You simply click the annotation and drag it within the visualization or to another visualization on the page.

Annotations can be hidden. You can either [hide them](#) totally or [minimize them](#) to icons.

### Procedure

1. On the menu bar, select **Visualizations > New annotation**.  
A text box with a toolbar opens on the active visualization.





2. Type your text in the box.  
As text is added, the box will increase vertically.




If you want to change the width of the box, place the cursor on the right edge of the box. When the cursor changes, click and drag the edge to resize the box.

3. On the toolbar, click the colored squares to change the background and text colors.

4. Click  or  to change the font size.  
The size of the annotation automatically adapts to the new font size.
5. Click outside the annotation to exit the input mode.  
The text annotation is added.



If you want to make changes to the annotation, simply click it. If you wish to delete the annotation completely, click it to access its toolbar and click Remove .

## Minimizing annotations

If an annotation covers important parts of the visualization, you can minimize it. A small icon will indicate the position of the annotation.


When the annotation is minimized, you can view its content in the tooltip. The tooltip is shown when moving the cursor over the icon.

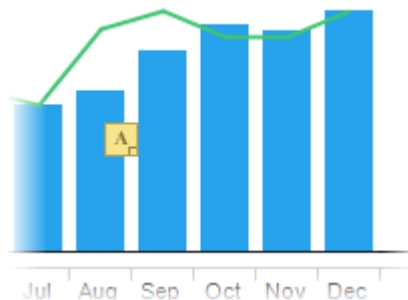
To view the full annotation, click the icon.

### Prerequisites

One or more annotations must be added to the visualization.

### Procedure

1. Place the cursor over the added annotation.  
The minimize button  is shown in the upper-left corner.
2. Click the button.  
The annotation is represented by an icon as shown in the image below.



To maximize the annotation, click the icon.

See also [Hiding annotations](#).

## Hiding annotations

Annotations are text boxes that can be added on top of a visualization. You can switch on and off the visibility of the annotations in a visualization.

### Prerequisites


One or more annotations must be added to the visualization.

### Procedure

1. Locate **Show/hide annotations**  in the visualization title bar.

2. Click **Show/hide annotations**.  
All annotations in the visualization are hidden.



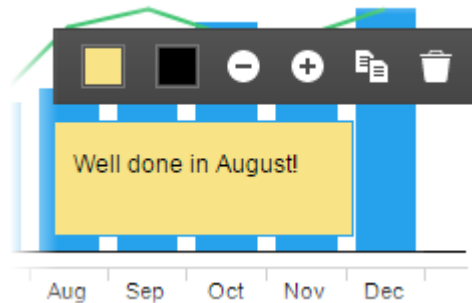
To show the annotations, click  in the visualization title once again.

## Removing annotations

Annotations are text boxes that can be added on top of a visualization. You can remove the annotations one by one.

### Procedure

1. Click on the annotation to access the annotation toolbar.



2. Click **Remove** .


## Duplicating an annotation

Annotations are text boxes that can be added on top of a visualization. You can duplicate the annotations in a visualization.

### Prerequisites


One or more annotations must be added to the visualization.

### Procedure

1. Click the annotation to access its toolbar.
2. On the toolbar, click **Duplicate** .
3. Click outside the annotations to exit the input mode.
4. Click and drag the created duplicate to the wanted position.


## Collaborating using conversations

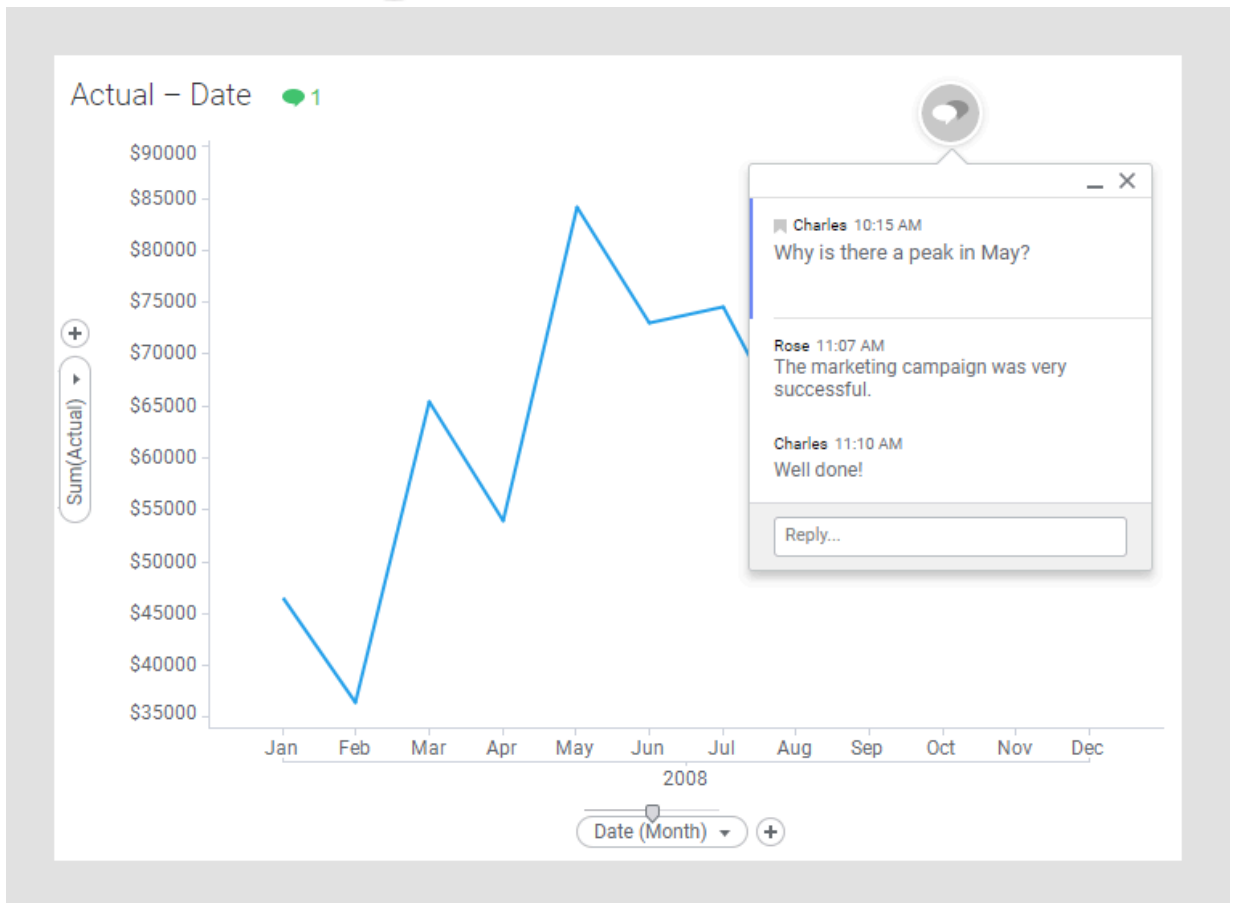
Collaboration with others on an analysis can be done by exchanging comments in a conversation. You start a conversation by inserting a comment, for example, to discuss what is visualized, or to ask for advice. The replies are then added in a sequence to the conversation thread.

The comments exchange is done in a specific mode that is switched on and off by clicking **Collaboration**  on the menu bar, or selecting **View > Collaboration**. When the mode is on, a floating toolbar is available.




You can add a comment to the visualization by clicking the green top button. Clicking the lower button in the floating toolbar opens a panel where you can browse conversations that already exist. You can answer any comment in the visualization and other people's replies can also be added to the conversation.

When the toolbar is open, the  symbol will be shown in the visualization.



Typically, the context in which a comment is made is relevant for what is communicated. Therefore, the posted comment contains not only your text message but it also captures the current [state of the analysis](#). The state includes, for example, on which position the comment is made, what is filtered and what is marked in the current view of the visualizations. Clicking the comment later on will immediately open the conversation in this captured context.

Replies that are posted in the thread can be pure text messages, or they can include another state of the analysis. Clicking a reply with an included analysis state will open the thread in this state of the analysis.

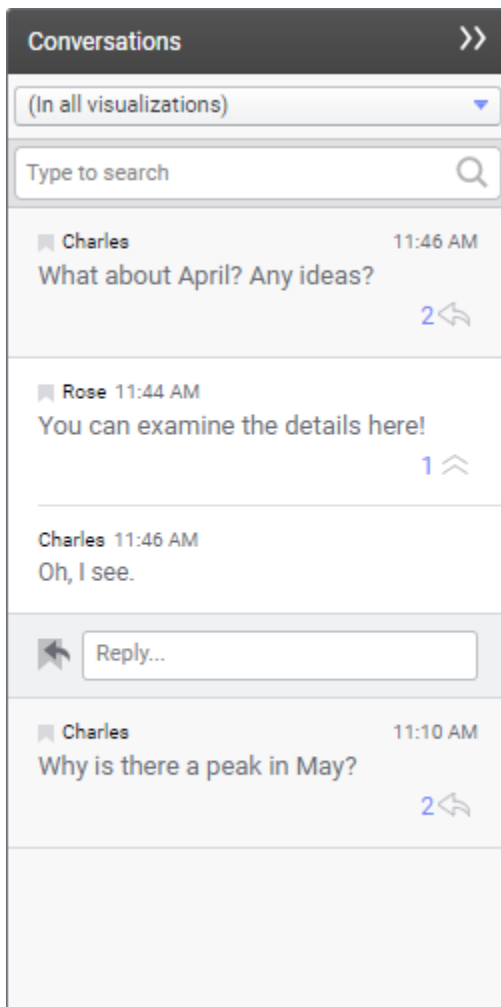
All comments that include captured analysis states are indicated by a  symbol, and will be opened in the analysis states, in which they were added.

A conversation thread can deal with anything from details in a particular visualization to the overall analysis. It will, however, reside in the visualization where the first comment was posted. If you move the visualization to another position, the thread will follow.

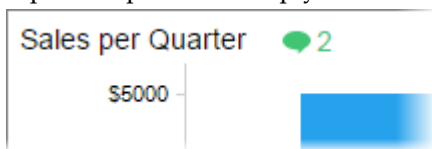
You can start as many conversation threads as you need. How many conversation threads that are related to a visualization is displayed in the visualization title. If you click the number, only these threads will be shown in the Conversations panel.

## Conversations panel



The Conversations panel lists the existing conversations in an analysis.



If you are interested in conversations related to a specific visualization, you can limit what is listed in the panel to these only. Select the visualization in question in the drop-down list at the top of the panel, or simply click the number displayed in the visualization title.



The conversations are listed in a chronological order with the conversation most recently updated at the top. The number of current replies is shown in the lower right of the initial comment.

From the panel you can choose to expand the full conversation within the panel by clicking  to expand it and  to collapse it. If you want to open the message in the context it was added, you click the initial comment. Replies can also be added from within the panel.

A search field is provided so you can look for specific conversations.

## Starting a conversation

You start a conversation in an analysis by posting a comment. The replies to the comment will be added to the conversation thread.

The posted comment contains your text message but it also captures the current [state of the analysis](#). This means that you, before posting the comment, can adjust the view to point to data of certain interest.

The comment is associated with the visualization in which it was posted. If you move the visualization to another position, the conversation thread will follow.

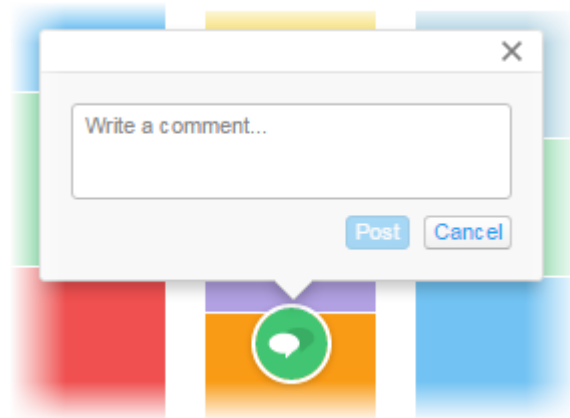
### Procedure

1. If the collaboration mode is not activated, click **Collaboration**  on the menu bar, or select **View > Collaboration**.
2. On the opened toolbar, click **Start conversation** .



Alternatively, you can replace both the steps above by right-clicking a visualization and selecting **Start conversation**.

A comment text box opens on the active visualization.



You can, if you want, move the comment by dragging the green anchor symbol to another position, within or outside the visualization.

3. Type your message in the text box.

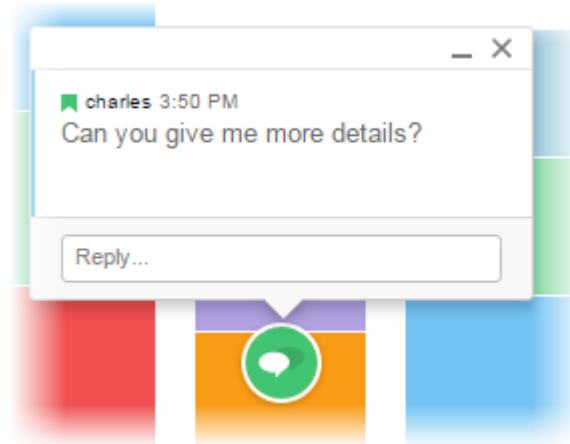


Filter to, and mark data of interest to draw attention to what you want to discuss in the analysis. The analysis state will be captured when posting the comment.



#### 4. Click **Post**.

The conversation is started and is now available to others for exchanging information. A reply can be added beneath the initial comment.



Moreover, in the visualization title, the number showing how many conversations are associated with the visualization increases by 1.

### Limitations in captured analysis states

The analysis state captured in a comment only includes the state for current marking, filtering and document properties. It does not include any visualization properties such as what columns you have on an axis, or if the bars are vertical or horizontal, or any other property you can change in the visualization properties. Moreover, it does not include changes in the underlying data.

The advantage of this approach is that comments do not lead to large changes in your analysis. You can therefore apply a comment, and from its included analysis state continue to analyze without having to fear unintended changes in the document. However, as a result of this approach, comments can become outdated if the analysis evolves.

### Replying in a conversation

You can reply to comments made by others, and your reply will be added to the conversation thread.

The reply is entered in the text field at the bottom of an [opened conversation](#). Typically you open a conversation in the context a comment was added, but it can also be shown in the Conversations panel.

#### Procedure

1. If the collaboration mode is not activated, click **Collaboration**  on the menu bar, or select **View > Collaboration**.

The Conversations panel lists existing conversations.



If you click the number in a visualization title, you can limit what is listed in the panel to conversations related to the visualization in question.

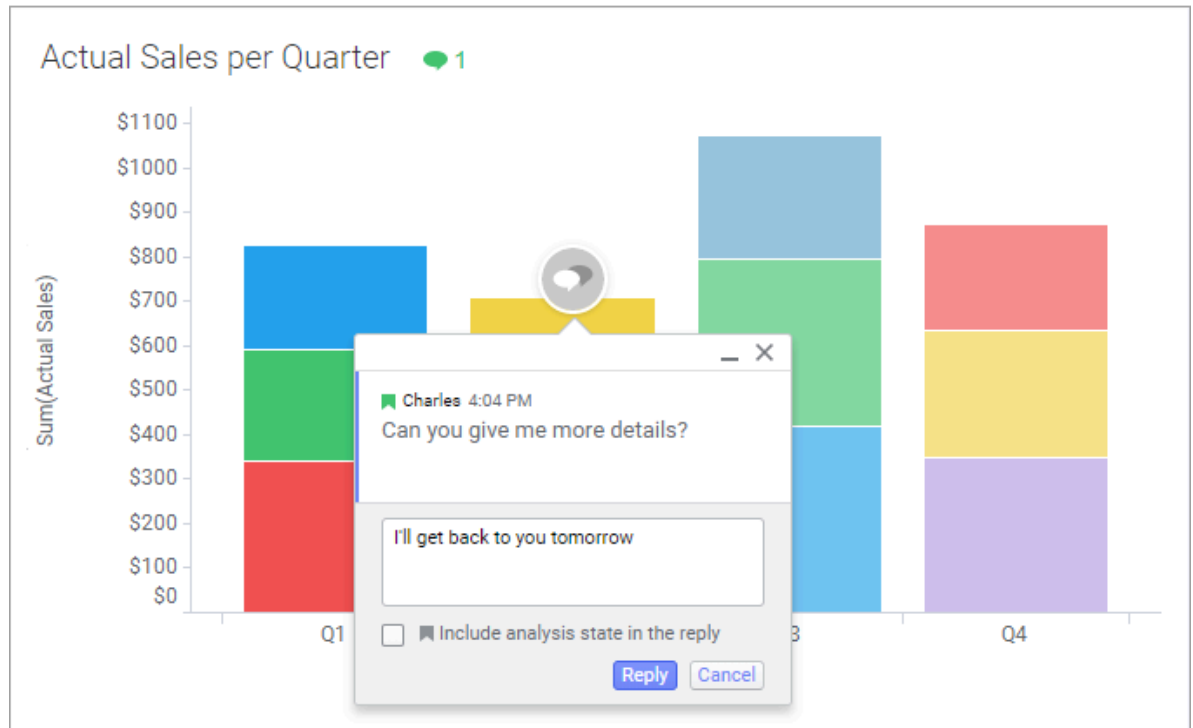
2. [Locate the conversation](#) to which you want to add a reply, and click it.

The conversation thread is opened in context, that is, it is opened in its associated analysis state.



Replies in the conversation can include captured analysis states. If you click such a reply, the conversation instead opens in the reply's associated state.

3. In the text box at the bottom of the thread, type your reply.



If you want the reply to be supplemented with an analysis state, follow the steps in the procedure below.

4. Click **Reply**.  
The reply is added to the thread.

### Replying with an analysis state in a conversation

A reply can contain text only, but if it is valuable to refer to a certain [state of the analysis](#), it is possible to supplement the reply with a captured state.


You can filter and mark values of your choice, move the conversation to another position, or add new visualizations to the analysis, and the analysis state can be included in the reply. Clicking such a reply later on will open the conversation in this captured state.

#### Procedure

1. Adjust the analysis to show the context you want to reference in the reply.



As soon as you click the text field, the conversation thread will also be shown on other pages. This means that you can open another page, adjust what is visualized, and include this current state of the analysis in your reply.

2. If you find it beneficial, drag the conversation to another position on the page.
3. Make sure that the check box **Include analysis state in the reply** is selected.
4. Click **Reply**.  
The reply is added to the thread and the included analysis state is indicated by a  symbol.

## Deleting a conversation



You can delete an entire conversation, that is, the comment initiating the thread together with all its replies.

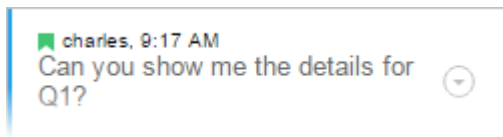
Conversations are handled in the collaboration mode, in which you can [access a conversation](#) in different ways.



The deletion of a conversation cannot be undone.

### Procedure

1. If the collaboration mode is not activated, click **Collaboration**  on the menu bar, or select **View > Collaboration**.
2. [Access the conversation](#) you want to delete.
3. Move the cursor over the first comment in the conversation, and click the **Menu** arrow .



4. Select **Delete**, and then click **OK** to confirm the deletion.



You can also [delete separate replies](#) in a conversation.

## Deleting a reply



A reply in a conversation can be deleted.

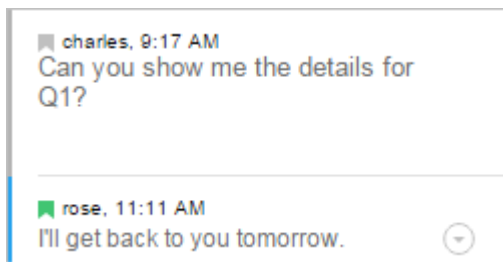
Conversations are handled in the collaboration mode. In this mode, you can [access a conversation](#) in different ways to delete one of its replies.



The deletion of a reply cannot be undone.

### Procedure

1. If the collaboration mode is not activated, click **Collaboration**  on the menu bar, or select **View > Collaboration**.
2. [Access the conversation](#) in which you want to delete a reply.
3. Move the cursor over the reply you want to delete, and click the **Menu** arrow .



4. Select **Delete**.




You can also [delete entire conversations](#).

## Viewing conversations

Depending on what you want to do, you can choose different ways of accessing and viewing conversations and their comments.


For example, you might be looking for an overview of which conversations have been held in the analysis, or you might want to view, or comment on, a specific conversation in its context.

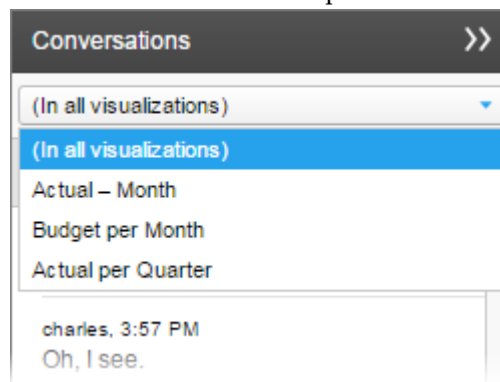


To access conversations, the collaboration mode must be turned on. If it is not activated, click **Collaboration**  on the menu bar, or select **View > Collaboration**.

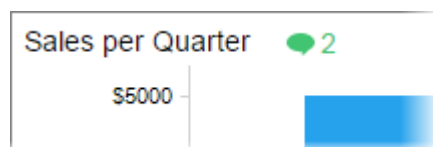
### View conversations in the Conversations panel



The Conversations panel provides an overview of existing conversations. You can

- list all conversations in the analysis
  - Click  on the collaboration toolbar, and all conversations are listed.
- list only conversations started within a specific visualization
  - Select the visualization in question in the drop-down list at the top of the panel.



- Alternatively, click the **Show conversations on this visualization** button in the visualization title.



In the panel, each conversation is represented by its initial comment. If you click  in the lower right of the initial comment, all replies in the conversation are shown (the current number of replies is indicated). Clicking  collapses the conversation thread again.

You can use the search field to locate conversations.

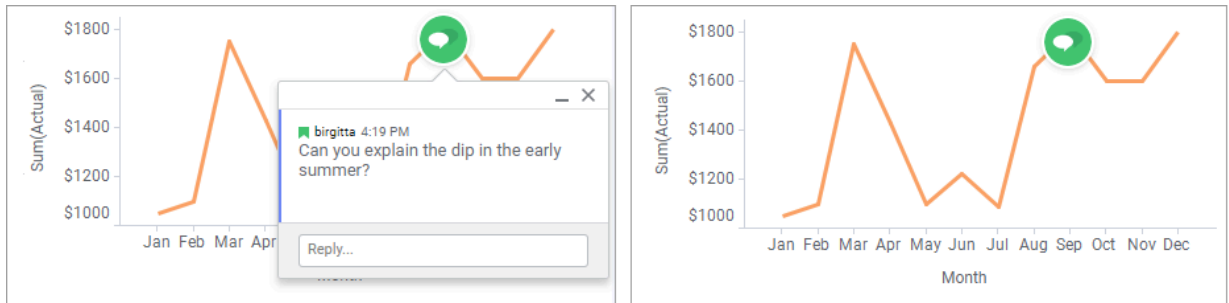
### View conversation in its context

Most of the times it is relevant to show the comments in a conversation in the context in which they were made. Anytime you click a conversation in the Conversations panel, the conversation opens in its associated analysis state, that is, in the context the first comment was posted.

When clicking a reply that includes an analysis state, the conversation opens in the state associated with the reply. This works not only when clicking replies in the Conversations panel but also when clicking replies in conversations displayed in context.

## Minimizing conversations


If a conversation covers parts that you want to view on a page, you can hide the comments in the thread. The anchor symbol will be kept.



Another option to make the interesting part visible is to move the conversation. Simply drag the anchor symbol to another position.

### Prerequisites



You must have a conversation shown on a page.

To minimize the conversation, click on the anchor symbol . The anchor symbol is still visible indicating the location of the conversation. To show the conversation, click the anchor symbol again.

## Getting link for sharing

In a conversation, you can get the links to comments including analysis states. The links can be shared with others in emails, blogs, or websites. Clicking the link will show the comment in its context, that is, in the analysis state it was posted.

### Procedure


1. If the collaboration mode is not activated, click **Collaboration**  on the menu bar, or select **View > Collaboration**.
2. **Locate the conversation** with the comment to share in the Conversations panel or in its context in the analysis.
3. Move the cursor over the comment or reply, and click the **Menu** arrow .
4. In the menu, select **Copy link**.
5. In the opened dialog, copy the link, and click **Close**.

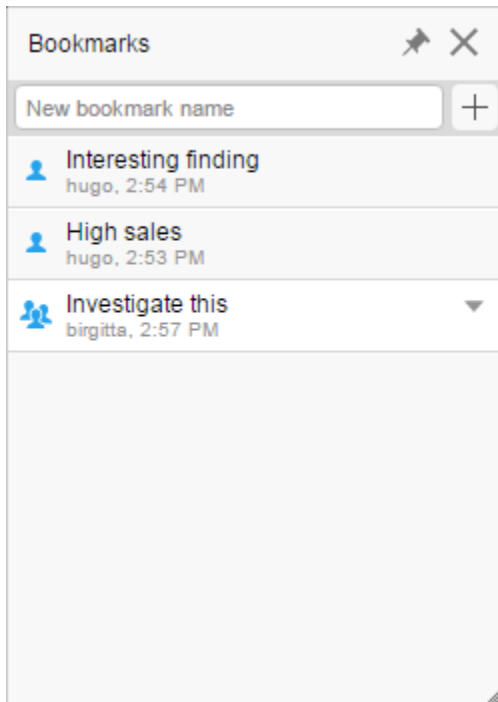
### What to do next

Paste the link into, for example, an email.

# Using bookmarks

While performing interactive analysis, you might want to capture analysis views with interesting findings to be able to revisit them later on. The use of bookmarks makes this possible because bookmarks are snapshots of the state of an analysis. Bookmarks can be shared with others, but they can also be kept private.

You can view and manage bookmarks in a popover or in a docked panel, opened by clicking  on the menu bar, or by selecting **View > Bookmarks**. The Bookmarks panel or popover will open in the mode it was opened the last time you had it open. The image below shows the Bookmarks popover with three added bookmarks. From the popover, you [add the bookmarks](#) as well as [apply them](#).



A bookmark is a snapshot of the current 'state' of an analysis, and it can be captured at any time. The state includes the following components:

- Page layout and visualization configuration, that is, selections on axes, coloring, formatting, and what is visible or hidden.
- Which items have been marked.
- Which page and visualization is active, including whether the active visualization is maximized.
- Filter settings.
- Filter organization (applicable to bookmarks created in the installed client).
- Properties and property values (applicable to bookmarks created in the installed client).

When a bookmark has been captured, you can, whenever you want, easily return to exactly what was captured, or share this state with others.





If you, after adding a bookmark, remove visualizations or pages, they will not be recreated when applying a bookmark. Neither will any pages or visualizations, which you have added after capturing a bookmark, be removed.



Spotfire attempts to find a symbolic representation of the data when a marking is applied; that means that all rows matching the marked category will be marked after data is reloaded. By using an aggregated visualization when marking, you can create a bookmark that marks all current rows matching the selected categories, rather than the absolute rows that were marked at the time of creation. See [Reapplying markings when reloading data](#) on page 532 for more information.

## Private and public bookmarks

There are [two kinds of bookmarks](#):

- Private bookmarks (indicated by ) , which only you can see. When adding a bookmark, it is private by default.
- Public bookmarks (indicated by ) , which are available to all users of the analysis.

In the popover/panel, private bookmarks are listed at the top and public bookmarks at the bottom.

## Bookmarks can be used as actions

When using the installed client, bookmarks can be included as links in a text area. This helps you create [guided analyses](#) where the recipient of your analysis can click on action links or buttons to quickly move through several different views of the analysis.

## Permissions

You can always apply any bookmarks that are visible to you in an analysis, but the ability to add and modify bookmarks can be restricted on two levels:

- Library folder permissions – The creator of an analysis can specify the folder permissions required to add bookmarks in the Document Properties dialog (installed client only). See [Details on Document Properties – Library](#) in the *Spotfire Analyst User Guide* for a list of the available alternatives. To learn more about folder permissions in the library, see [Permissions](#) in the *Spotfire Analyst User Guide*.
- Licenses – Which [license](#) you have determines whether or not you will be able to add bookmarks to analyses. Contact your administrator if you are missing a license feature.

## Limitations on Bookmarks



- It might not be possible to apply all parts of a bookmark if there are significant changes to the underlying data or the analysis. If the bookmark cannot be applied, you will see a notification about the issues.
- If data is refreshed, markings in unaggregated visualizations can only be reapplied provided that key columns have been configured for that data table. See also [Reapplying markings when reloading data](#) on page 532.
- Bookmarks are saved per user, per document. If your web client is configured for “impersonation” to allow multiple users to log in anonymously, these users all impersonate a single user profile, so any private bookmarks captured by one user will be visible to all the other users under the same user profile.
- Bookmarks cannot capture filtering which is created using **Marked rows > Filter to** or **Filter out**.

## Adding a bookmark

By adding a bookmark to an analysis, you can capture an interesting view in the analysis. You, and if you want also your colleagues, can then revisit it instantly by clicking the added bookmark.

Bookmarks are managed in a popover or in a docked panel. It will open in the mode it was opened the last time you used it.

### Procedure

1. Open Bookmarks by clicking  on the menu bar, or by selecting **View > Bookmarks**.
2. In the text field, enter a name for the bookmark.
3. Click **Add bookmark**, .



In the installed client, you can also press CTRL+B on the keyboard to add a bookmark, or press CTRL+Shift+B (or use right-click on the Add bookmark button), to open the Add Bookmark Special dialog, where you can select to add partial bookmarks.

The different parts of a bookmark are described below.

Option	Description
<b>Page Layout and Visualizations</b>	Captures the layout and the configuration of the visualizations on the active page. This includes all specified visualization properties. For example, selections on the axes of a visualization, coloring, formatting, and which visualization features are shown or hidden.
<b>Active Page</b>	Captures which page is active when the bookmark is created.
<b>Active Visualization</b>	Captures which visualization is active when the bookmark is saved. If the active visualization was maximized, it is captured in this bookmark part.
<b>Filter Settings</b>	Captures the filtering and filter types used on the active page.
<b>Filter Organization</b>	Captures the layout of the filters panel (sort order, filter groups, hidden filters).
<b>Markings</b>	Captures which items are marked in the visualizations in the bookmark. Note that the active marking for a visualization is included in the Page Layout and Visualizations bookmark part.
<b>Properties</b>	Captures any editable document, data table, and column properties used on the active page, as well as any property values referred to in a property control.

### Result

The bookmark is added to the list in Bookmarks. By default, the bookmark is **private**.



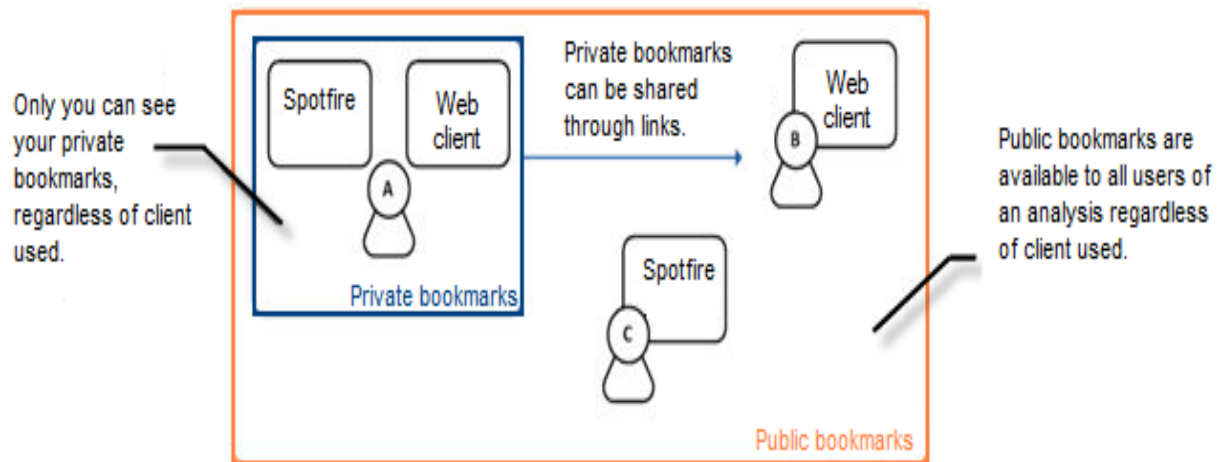
You can **update** a bookmark at a later stage.

## Bookmarks example scenarios

The concept of Bookmarks has many potential usages and usage goals. This topic presents some possible scenarios where bookmarks can be useful.

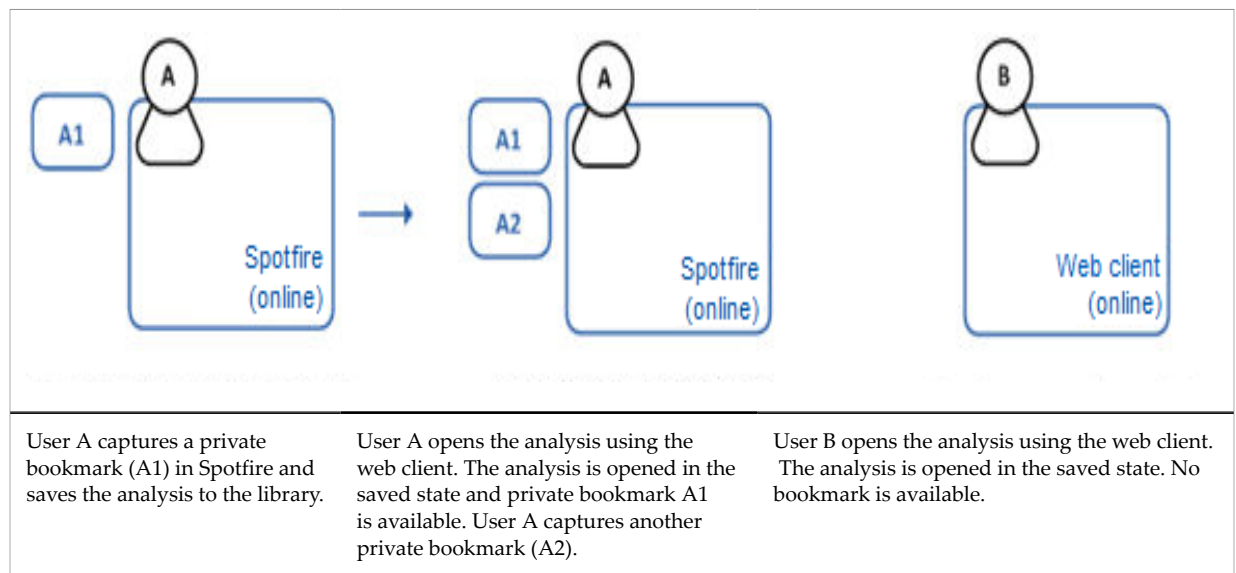
Where only "Spotfire" is mentioned in the images, we assume that the installed client is used.





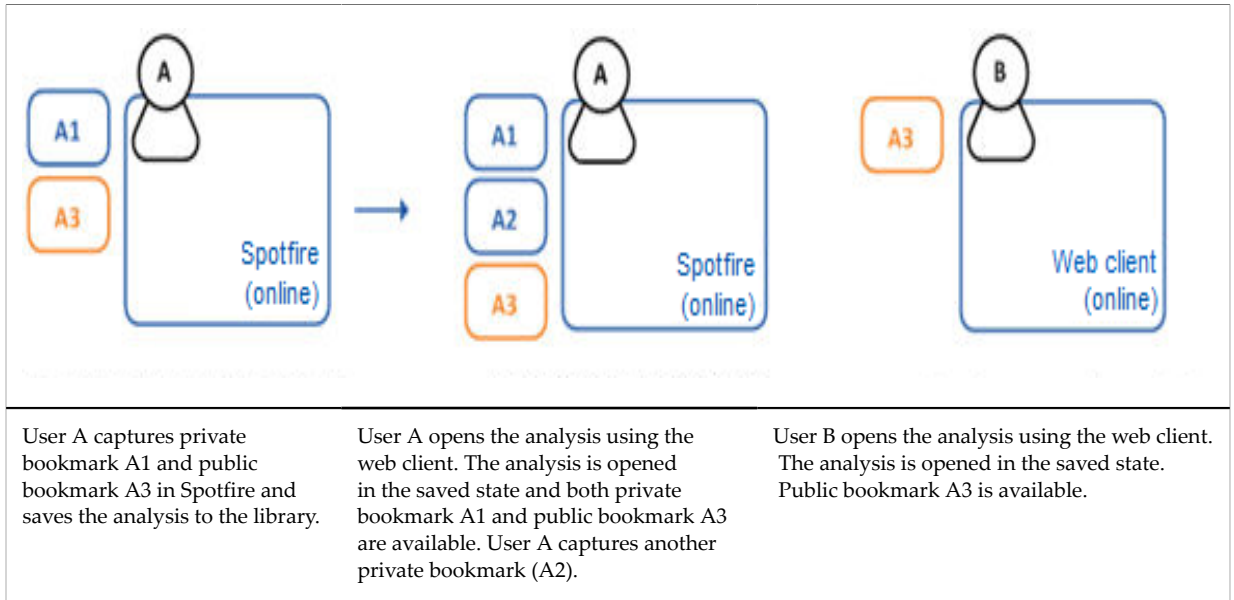
The example flows in this topic aim to describe how private and public bookmarks work across clients and between different users.

### Private bookmarks - available to a user regardless of client used



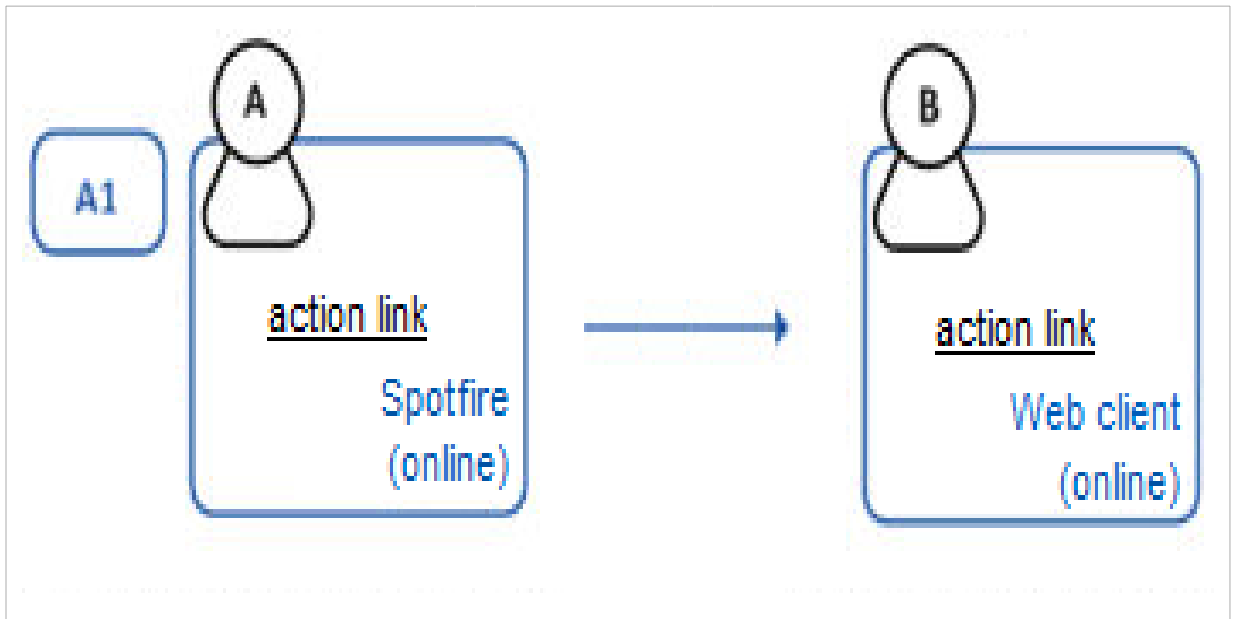
Private bookmarks can be used to save settings (an analysis state) from one occasion to another, to avoid repetitive work every time you open up an analysis. You can also use private bookmarks to capture an interesting state that later can be revisited, to analyze previously captured insights by sharing states with others or comparing them with other states, to understand how different parameters (settings) affect the result.

### Public bookmarks - available to all users of an analysis regardless of client used



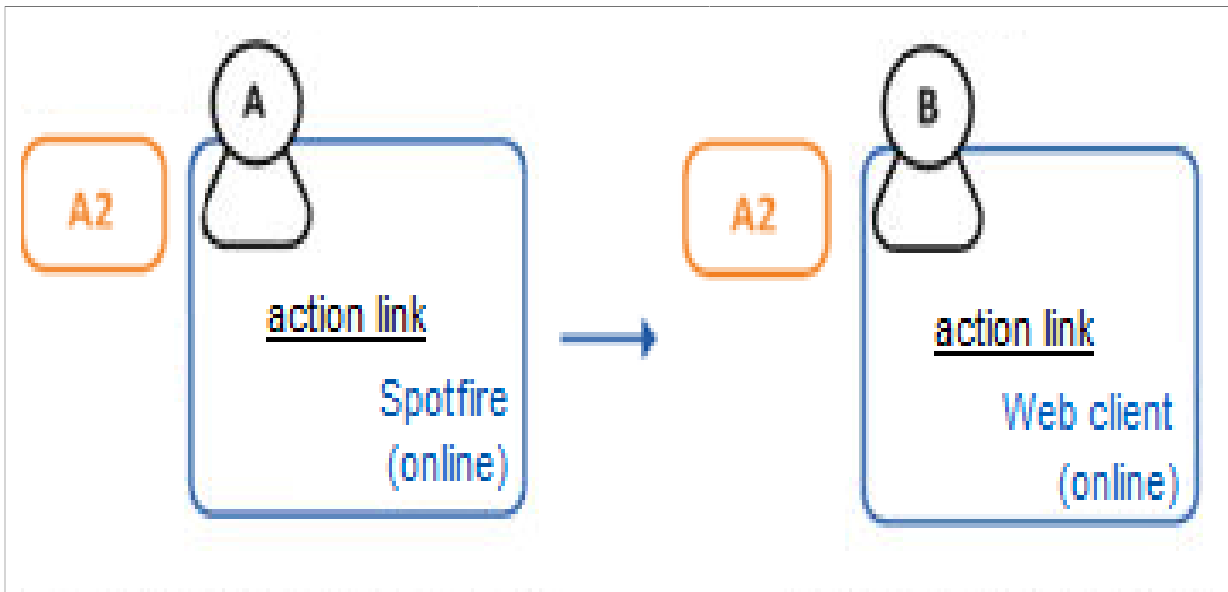
Public bookmarks can be used as starting points for end users of an analysis. The captured states can guide users to interesting aspects in an analysis, or they might be used for navigation as an alternative to pages or links. By sharing captured states with other users you can also induce discussions of the insights made during analysis.

### Private and public bookmarks as building blocks in action links



User A captures a private bookmark (A1) in Spotfire and ties it to an action link, then saves the analysis to the library.

User B opens the analysis using the web client. The action link sets the bookmark state. No bookmark is available to user B.

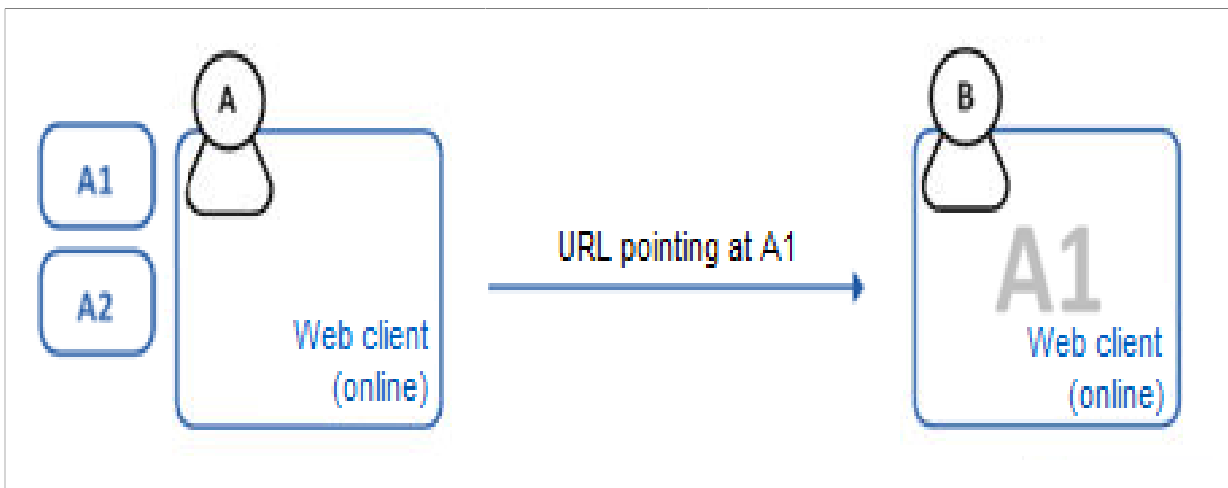


User A captures a public bookmark (A2) in Spotfire and ties it to an action link, then saves the analysis to the library.

User B opens the analysis using the web client. The action link sets the bookmark state. Bookmark A2 is available.

Both private and public bookmarks can be used as building blocks in action links/buttons when setting up a guided analysis for other users. The action links/buttons typically guide users to insights or relevant starting points in the analysis. This also enables a way to include some restricted interaction possibilities in the application and a possibility to hide controls such as the filters panel.

### Send private bookmark to colleague

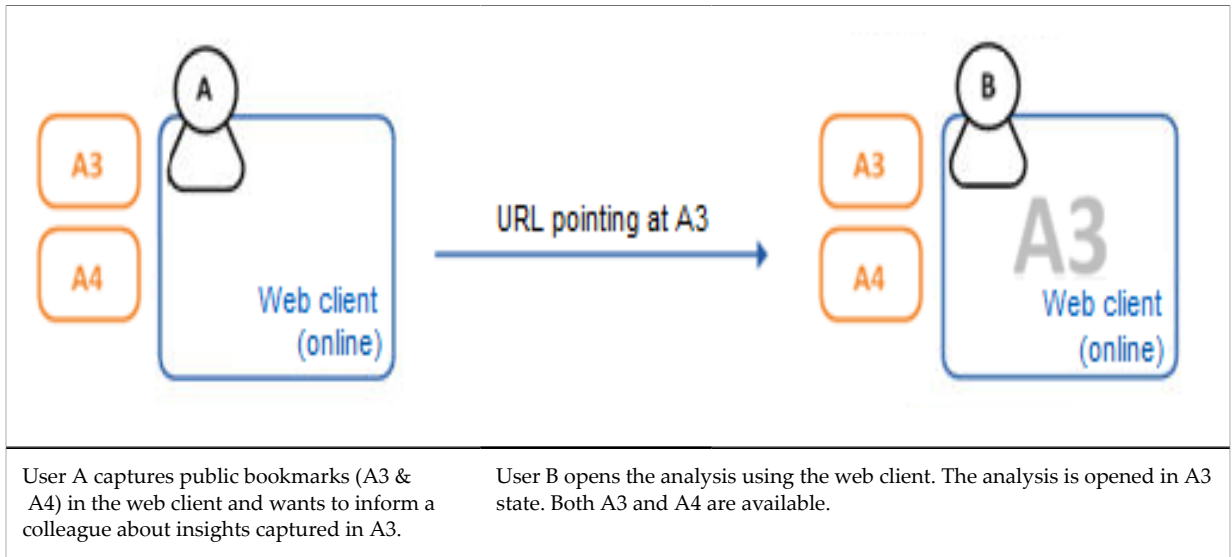


User A captures private bookmarks (A1 & A2) in the web client and wants to inform a colleague about insights captured in A1.

User B clicks a link in an email, which opens the analysis in the web client. The analysis is opened in A1 state. No bookmark is available.

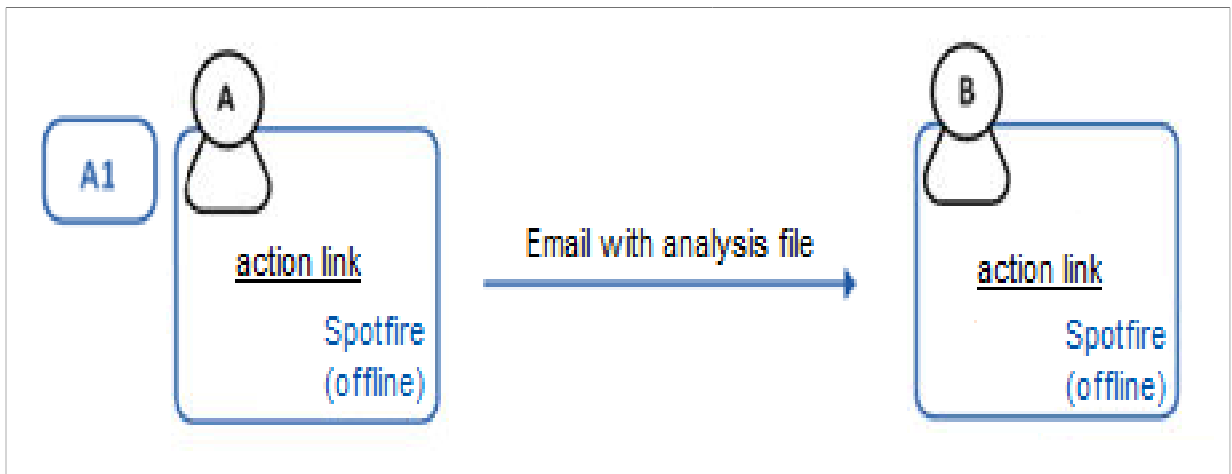
A private bookmark can be used to pass on a captured insight to a specific colleague or group of colleagues only. This can easily be done by sending a URL that points to an analysis in a particular (bookmark) state. See [Links to analyses in the library](#) on page 989 for more information about different link types.

### Send public bookmark to colleague



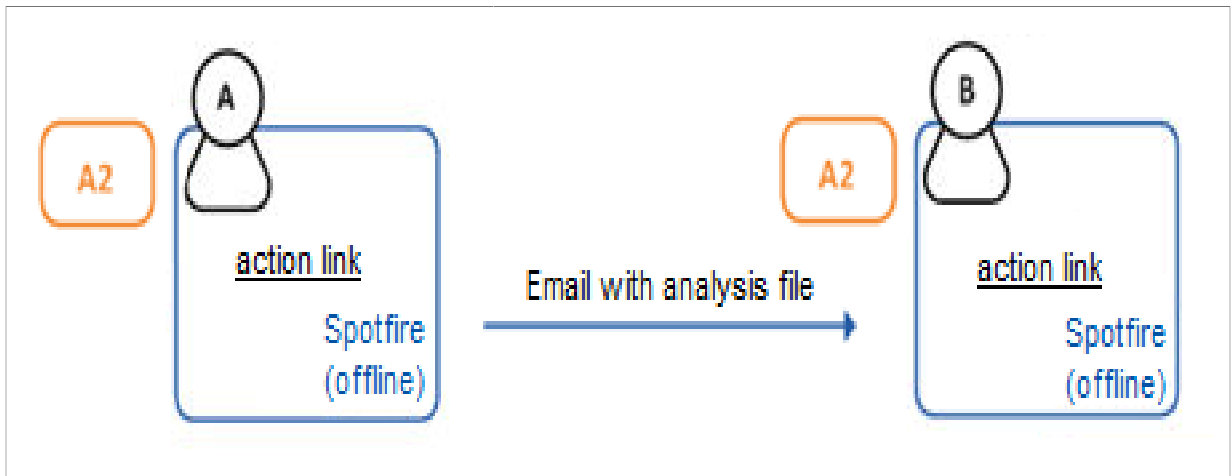
Public bookmarks can also be referred to directly using a link.

### Offline collaboration



User A captures a private bookmark (A1) in Spotfire and ties it to an action link, then saves the file and sends it to a colleague.

User B opens the analysis in Spotfire. The action link sets the bookmark state. No bookmark is available to user B so A1 cannot be updated by user B.



User A captures a public bookmark (A2) in Spotfire and ties it to an action link, then saves the file and sends it to a colleague.


User B opens the analysis in Spotfire. The action link sets the bookmark state. Bookmark A2 is available.

## Applying a bookmark

A bookmark is a captured snapshot of interesting findings in an analysis. You can revisit what is captured anytime by applying the bookmark.

Bookmarks are managed in a popover or in a docked panel. It will open in the mode it was opened the last time you used it.

### Procedure

1. Open Bookmarks by clicking  on the menu bar, or by selecting **View > Bookmarks**.
2. Click the bookmark to apply it.



If you want to apply only a specific part of the captured bookmark, you can click the arrow next to the bookmark and select **Apply special**. In the opened menu, select the part of interest. See [Adding a bookmark](#) on page 700 for more information about different parts.



If you apply filter settings in a bookmark using **Apply special** from the installed client, you can select whether to **Set All Filters** (which applies the exact filter settings for the active filtering scheme, as stored in the bookmark), **Set Only Adjusted Filters** (which updates only those filters that were changed in the bookmark), or **Mark Filtered Rows** (which uses the filter condition from the bookmark to mark rows in the analysis but leaves the filtering as it was before applying the bookmark).



Using the installed client, bookmarks can also be added to [action controls](#) (links, buttons or images) in a text area or as [actions performed when clicking on visualization items](#). In that case, clicking on the action control will apply the bookmark.


## Updating a bookmark

---

You can update what is captured in a bookmark, if you find a better view describing the interesting finding.

Bookmarks are managed in a popover or in a docked panel. It will open in the mode it was opened the last time you used it.

### Procedure

1. Make sure what you currently see in the analysis is what you want the bookmark to capture.
2. Open Bookmarks by clicking  on the menu bar, or by selecting **View > Bookmarks**.
3. Move the cursor over the bookmark you want to update, and click the menu arrow that is displayed.
4. In the opened menu, select **Update**.  
What is captured in the bookmark is replaced by the current view of the analysis.


## Renaming a bookmark

---

You can change the name of the bookmark.

Bookmarks are managed in a popover or in a docked panel. It will open in the mode it was opened the last time you used it.

### Procedure

1. Open Bookmarks by clicking  on the menu bar, or by selecting **View > Bookmarks**.
2. Move the cursor over the bookmark you want to rename, and click the menu arrow that is displayed.
3. In the opened menu, select **Rename**.  
A text field is displayed.
4. Type the new bookmark name.
5. Click **Save**.  
The new bookmark name replaces the previous name.


## Deleting a bookmark

---

If a bookmark is no longer relevant, you can delete it.

Bookmarks are managed in a popover or in a docked panel. It will open in the mode it was opened the last time you used it.

### Procedure

1. Open Bookmarks by clicking  on the menu bar, or by selecting **View > Bookmarks**.
2. Move the cursor over the bookmark you want to delete, and click the menu arrow that is displayed.
3. In the opened menu, select **Delete**.  
The bookmark is removed from the analysis.


## Making a bookmark public

---

Private bookmarks are only available to you, and public bookmarks are available to all users. When you add a bookmark, it is private by default. However, you can make a private bookmark public, and vice versa.

Bookmarks are managed in a popover or in a docked panel. It will open in the mode it was opened the last time you used it.

### Procedure

1. Open Bookmarks by clicking  on the menu bar, or by selecting **View > Bookmarks**.
2. Move the cursor over the bookmark you want to make public, and click the menu arrow that is displayed.
3. In the opened menu, select **Public bookmark**.

The bookmark gets visible in Bookmarks, and it can be applied by every user.



It is possible to share also a private bookmark with others. [Send them the URL to the private bookmark](#), for example, in an email.


## Making a bookmark private

---

Public bookmarks are available to all users, and private bookmarks are only available to you. You can make a public bookmark private, and vice versa.

Bookmarks are managed in a popover or in a docked panel. It will open in the mode it was opened the last time you used it.

### Procedure

1. Open Bookmarks by clicking  on the menu bar, or by selecting **View > Bookmarks**.
2. Move the cursor over the bookmark you want to make private, and click the menu arrow that is displayed.
3. In the opened menu, select **Private bookmark**.

The bookmark is only available to you.



It is possible to share private bookmarks with others. [Send them the URL to the private bookmark](#), for example, in an email.

## Copying a link to a bookmark

---

You can copy URLs pointing to analysis views that are captured in bookmarks and then share them with others, if the analysis is saved in the library.


Bookmarks are managed in a popover or in a docked panel. It will open in the mode it was opened the last time you used it.



If a [bookmark is private](#), you can still send a URL including the private bookmark to a colleague to share that bookmark.

### Procedure

1. Make sure the analysis is saved in the library.

2. Open Bookmarks by clicking  on the menu bar, or by selecting **View > Bookmarks**.
3. Move the cursor over the bookmark in question, and click the menu arrow that is displayed.
4. In the opened menu, select **Copy bookmark URL**.
  - In the installed client, you can directly select which type of link to copy from the menu. Read more about different types of links in [Links to analyses in the library](#) on page 989.
  - In the web client, the **Copy link** dialog is displayed, and you can select and copy the link from the **Link to bookmark** field. Click **Close** when you are done. Links copied from the web client always assume that you want to open the link with a web client.

## Result

You can now share the link to the bookmark with someone else, for example by pasting the link into an email.



## Grouping data

---

When working with an analysis, it might be beneficial to group the data in various ways to serve different purposes.

- [Tags](#)

You can group data rows exactly as you wish by attaching tags to data rows.

- [Creating intervals of numerical data](#)

You can group numerical data into intervals that are then handled as categories.

- [Grouping categories](#)

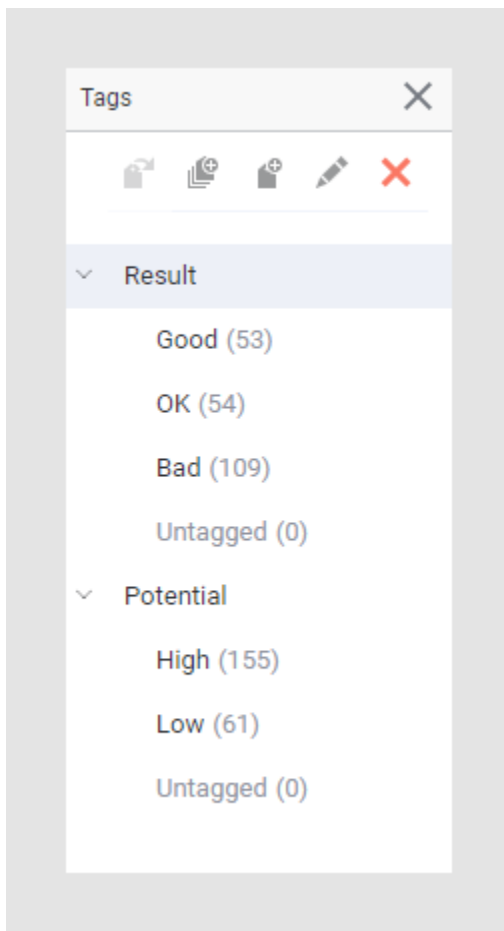
You can group categorical values into a new categorical value. This can be useful when a data column contains categorical values that are not important to display separately.

## Tags

---

The attachment of tags to data rows is a method to distinguish and group rows exactly as you wish. It could be any data table rows that you consider have something in common, and you would find it beneficial to be able to identify them quickly in different visualization contexts. For example, you might want to assign a 'Top 10' tag to the highest ten values of a measure, or tag rows with 'Good', 'OK', or 'Bad'. Rows with the same tag, and also the rows that are untagged, can then easily be identified in a visualization.

You can view and handle tags in a panel, opened by selecting **View > Tags** on the menu bar. The image below shows an example of the Tags panel. Once rows have been tagged, you simply double-click a tag in the panel to mark all its rows in the visualizations.

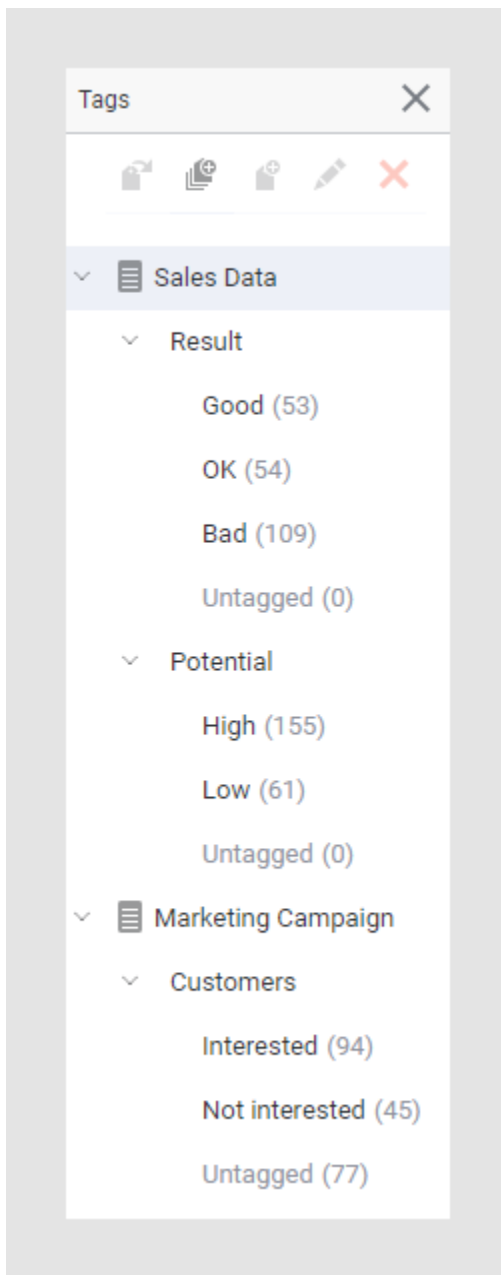


The tags used to group the data rows form a tag collection, and the tag collection becomes a new data column in the data table. Its different tags are the column values. The column is treated in the same way as any other column.



It is possible to create more than one tag collection within a data table as shown above (the tag collections 'Result' and 'Potential').

You attach the tags to rows that have been [marked](#). The active visualization defines which marking to get the marked rows from. Each row can only contain a single tag from each tag collection. Tags can only be attached to rows from a single data table, but the same tag collection and tag names can be used for multiple data tables.



If tags are to be reapplied after reloading linked data, you need to specify key columns that can be used to uniquely identify the rows in each data table. This can be done using the installed client.



Tags are not supported when working with in-database data.



## Creating tags

You might want to group certain data rows to be able to examine them in various contexts.

By attaching [tags](#) to these rows, they can be easy to spot. In the Tags panel, you first create a tag collection, which you add tags to.

### Procedure

1. If the Tags panel is not already visible, select **View > Tags** on the menu bar.

2. If you have more than one data table in the analysis, select the data table where you want to add the tag collection.
3. Click **New tag collection**, .
4. In the New tag collection dialog, type a **Name** for the tag collection.  
This name will also be used as a column name, when the tag collection is viewed as a column in visualizations.
5. Optionally, type a **Description** with details about the tag collection.
6. To add a tag, click **New**.
7. Type a **Tag name**, and click **OK**.  
The tag is added below **Tags** in the **New tag collection** dialog.
8. Repeat steps 6 and 7 until you have added all tags that you want to include in the tag collection, and click **OK**.
9. In the visualization, mark the rows that you wish to assign a certain tag. In the Tags panel, then select this tag and click **Attach tag to marked rows**, .  
Alternatively, drag the desired tag to the center of the visualization, and drop it on the Attach tag drop target. You can only attach one tag at a time.
10. To find all rows with a certain tag, double-click the tag in the Tags panel.  
All rows with the attached tag get marked in the visualizations.



You can distinguish all rows without a tag by double-clicking **Untagged** in the Tags panel.

## Removing tags

You can remove tags that have been attached to rows.

### Remove all the tags in a specific tag collection from rows

#### Procedure

1. If the Tags panel is not already visible, select **View > Tags**.
2. For the desired data table, right-click the tag collection.
3. From the opened pop-up menu, select **Remove all tags**.  
A confirmation message is displayed.
4. Click **OK**.  
All tags are removed for that specific tag collection (in the specific data table).

### Remove all tags from marked rows

#### Procedure

1. In the visualization, mark the rows that you want to remove tags from.
2. If the Tags panel is not already visible, select **View > Tags**.
3. In the desired data table and tag collection, right-click **Untagged**.
4. From the opened pop-up menu, select **Remove tags from marked rows**.  
All tags from the specified tag collection are removed from the marked rows. Alternatively, drag the **Untagged** item in the tag collection to the visualization and drop it on the drop target.

## Remove a single tag from all rows

### Procedure

1. If the Tags panel is not already visible, select **View > Tags**.
2. For the desired data table, right-click the tag you want to remove from all rows.  
If you have more than one data table in your analysis, and the same tag is available in more than one data table, you need to make sure that you right-click the tag in the actual data table of interest.
3. From the opened pop-up menu, select **Remove tag from all rows**.  
The selected tag is removed from all rows in this data table.

## Using tags - example

This example is intended to illustrate a use case of tags.

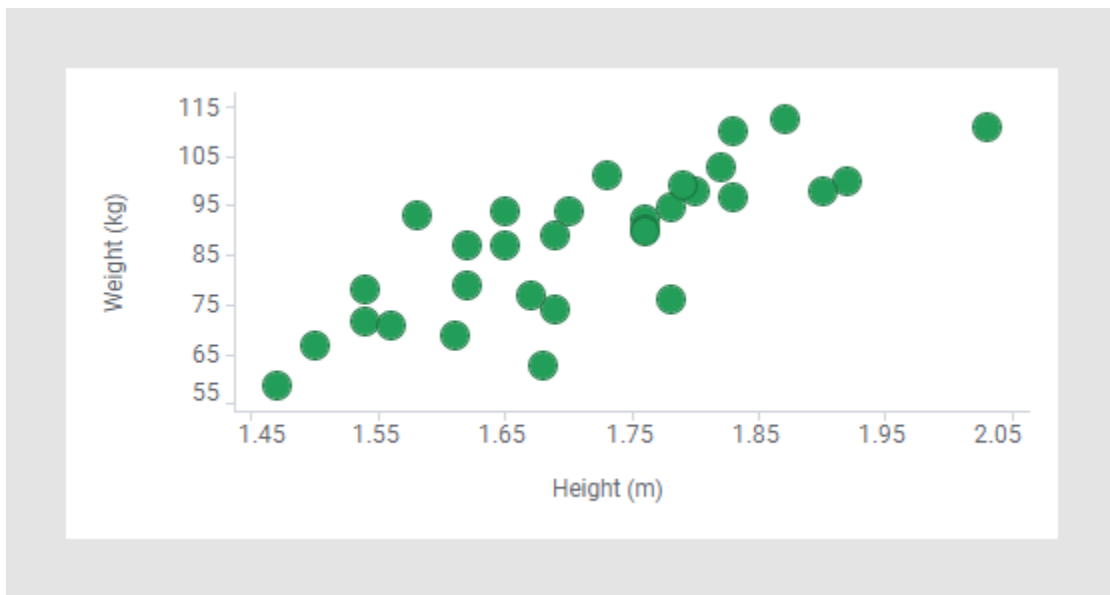



Tags are not supported when working with in-database data.

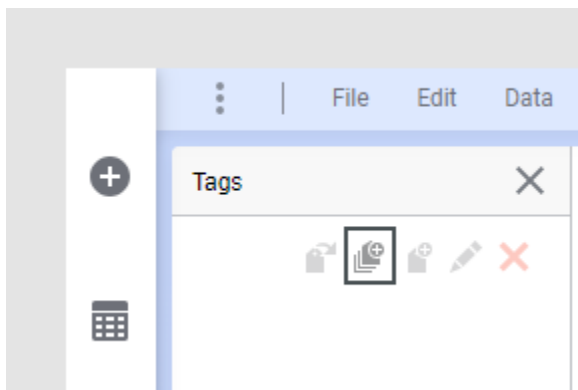
In a health investigation, Height (m), Weight (kg), and BMI (Body Mass Index) are registered for a number of people in the data table below.

Gender	Height (m)	Weight (kg)	BMI
female	1.65	87.10	31.99
female	1.50	67.00	29.78
female	1.69	74.20	25.98
male	1.58	93.20	37.33
male	1.76	92.40	29.83
male	1.82	102.80	31.03
male	1.92	99.80	27.07
female	1.54	71.60	30.19
female	1.76	90.40	29.18
female	1.67	76.80	27.54

Assume it is of interest to distinguish individuals with a BMI value higher than 30, and individuals with a BMI value lower than 30, for example, in a scatter plot, where the weights are plotted against the heights. You can do that by attaching different tags to the data rows.



First, create a tag collection. Select **View > Tags** on the menu bar to open the Tags panel, and click **New tag collection**, .



In the New tag collection dialog, give the tag collection a **Name**, and create the tags via **New**.

**New tag collection**

Name

BMI groups

Description

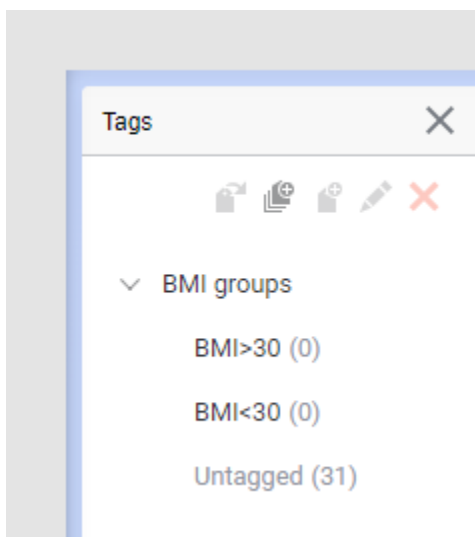
Tags


BMI>30

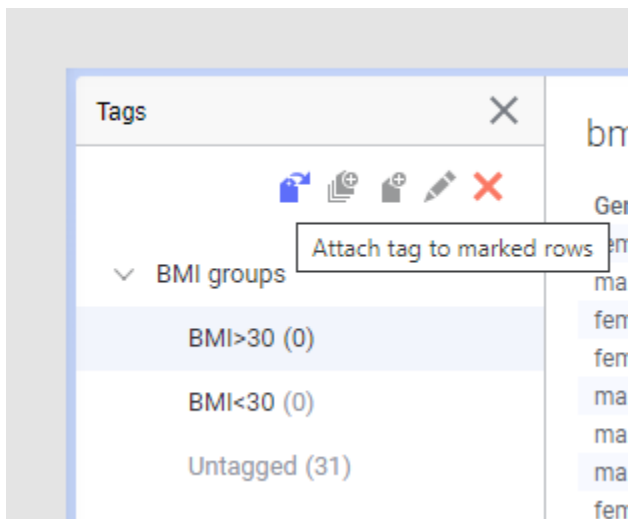
BMI<30

New

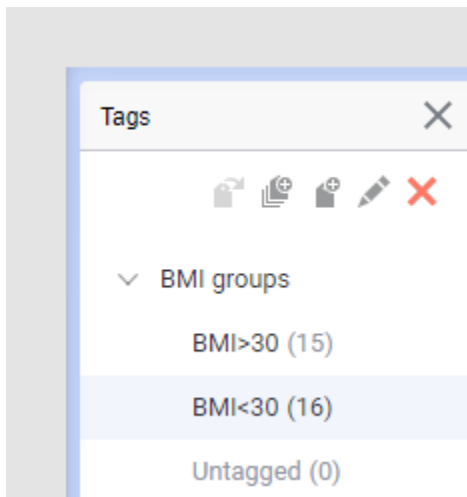
The Tags panel now shows the tag collection and its tags.



No tags have yet been attached to any rows, as indicated by the zeros in the parentheses. To attach tags, locate the rows you want to attach a tag to, for example, by sorting the BMI column in the table, and mark the rows with a BMI value above 30. Then select the BMI>30 tag in the Tags panel, and click **Attach tag to marked rows**, . In the same way, attach the BMI<30 tag to the remaining rows.

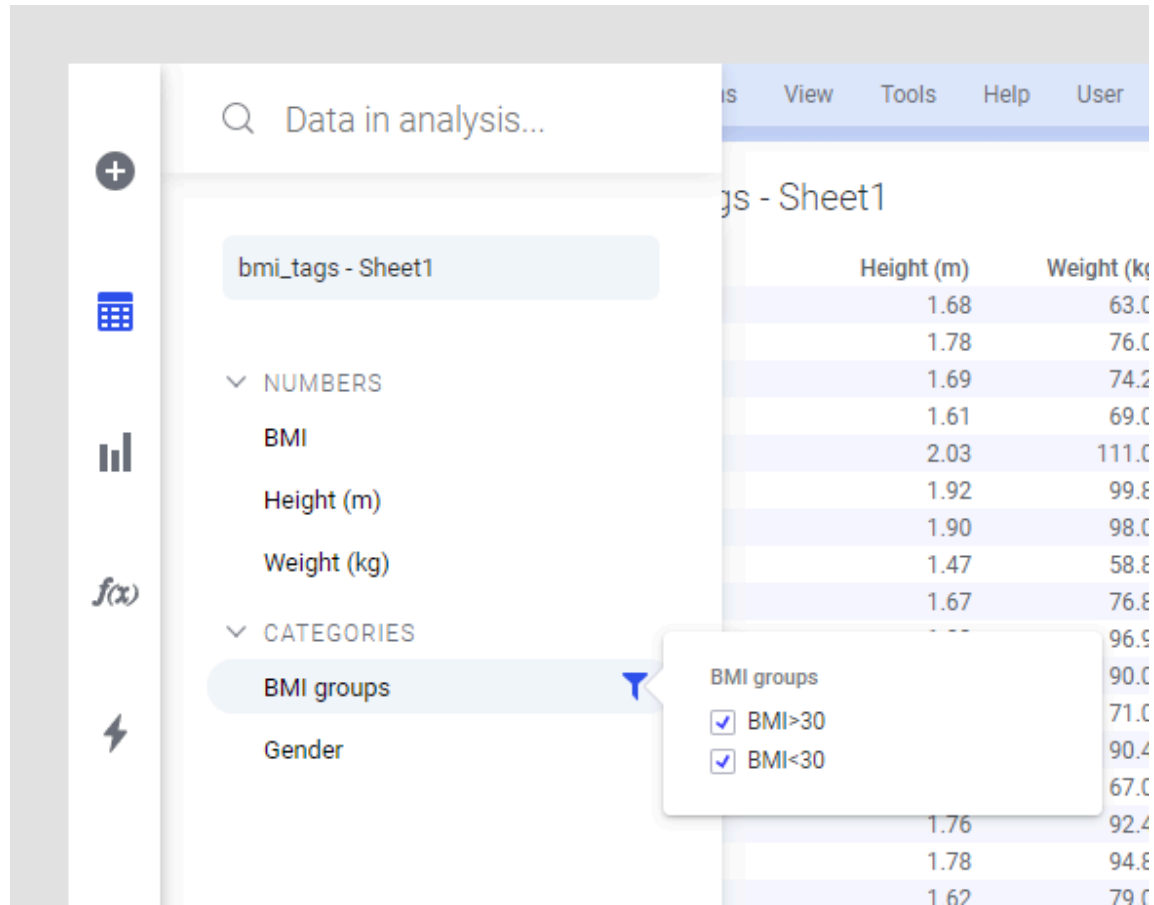


The number of rows that have been attached with a certain tag is shown in the parentheses.





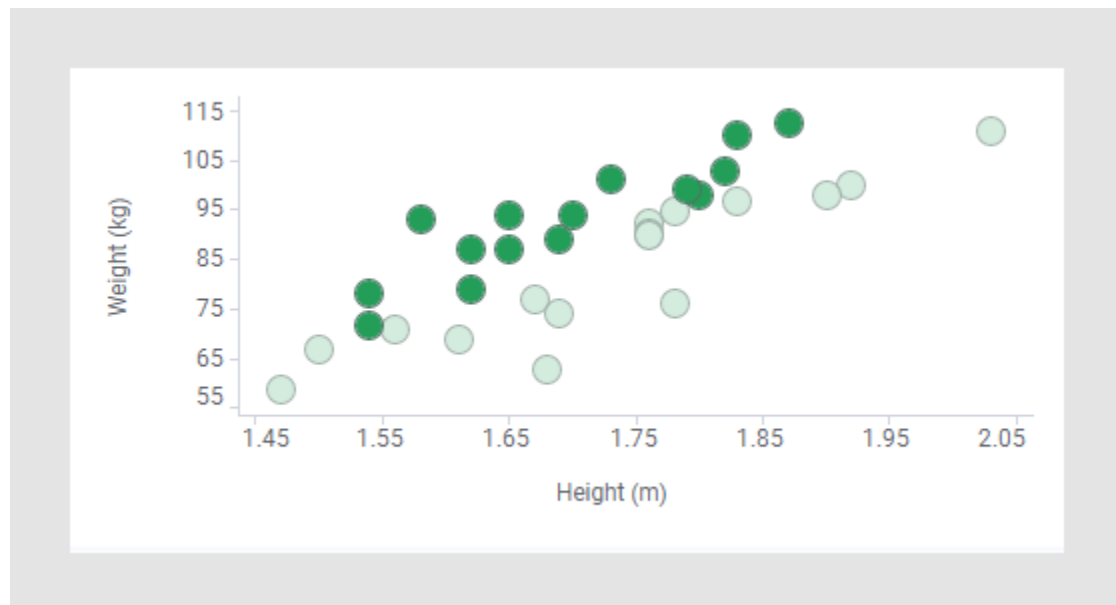
The created tag collection results in a new data column, and a new filter, as shown below.



The screenshot shows the Spotfire Web Client interface. On the left is a sidebar with icons for adding data, viewing data, and creating visualizations. The main panel displays a search bar 'Data in analysis...' and a list of data sources. The selected data source is 'bmi\_tags - Sheet1'. Below this, there are two sections: 'NUMBERS' and 'CATEGORIES'. Under 'NUMBERS', there are three items: 'BMI', 'Height (m)', and 'Weight (kg)'. Under 'CATEGORIES', there are two items: 'BMI groups' and 'Gender'. A dropdown menu is open for 'BMI groups', showing two options: 'BMI>30' and 'BMI<30', both with checked checkboxes. In the background, a table of data is visible with columns 'Height (m)' and 'Weight (kg)'. The table contains 10 rows of data.

Height (m)	Weight (kg)
1.68	63.0
1.78	76.0
1.69	74.2
1.61	69.0
2.03	111.0
1.92	99.8
1.90	98.0
1.47	58.8
1.67	76.8
1.76	92.4
1.78	94.8
1.62	79.0

Once tags have been attached, you simply double-click a tag in the Tags panel, and its associated rows become marked in the visualizations. For example, when you double-click the BMI>30 tag, the following individuals have a BMI higher than 30 in the scatter plot.

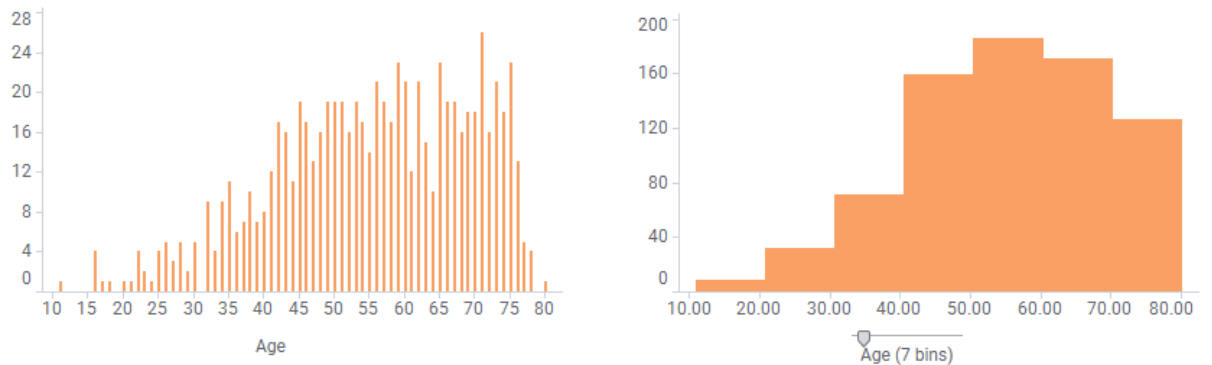


## Creating intervals of numerical data

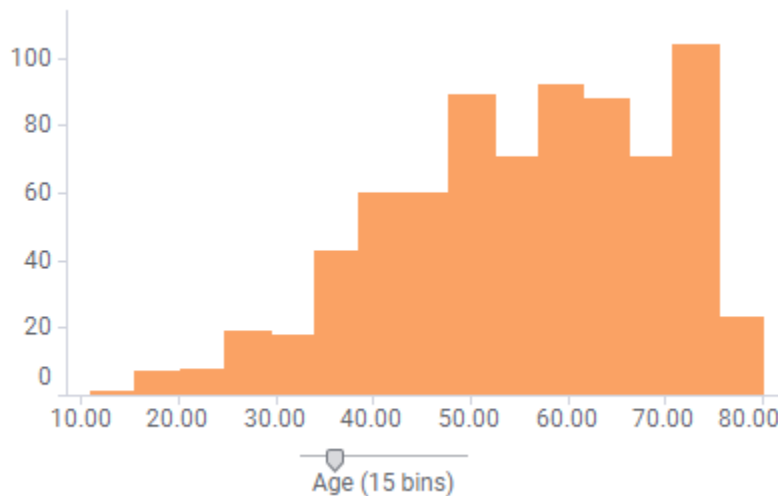
Numerical data can be grouped into intervals. The intervals are named bins and can be handled as categories in an analysis.

The values in a numerical column may not be important individually. Sometimes bins of values are preferred because they can be displayed as categories in a visualization. When you create bins, the range between the lowest value and the highest value in the numerical data column is divided into a number of intervals.

For example, if your data contains age information about a group of people, you may want to arrange the ages into a smaller number of age groups as illustrated below.



You decide how many bins the range should be divided into. When binning the data, a slider appears that can be dragged to the wanted number of bins. The bar chart below is based on the same data as above but adjusted to display more bins.




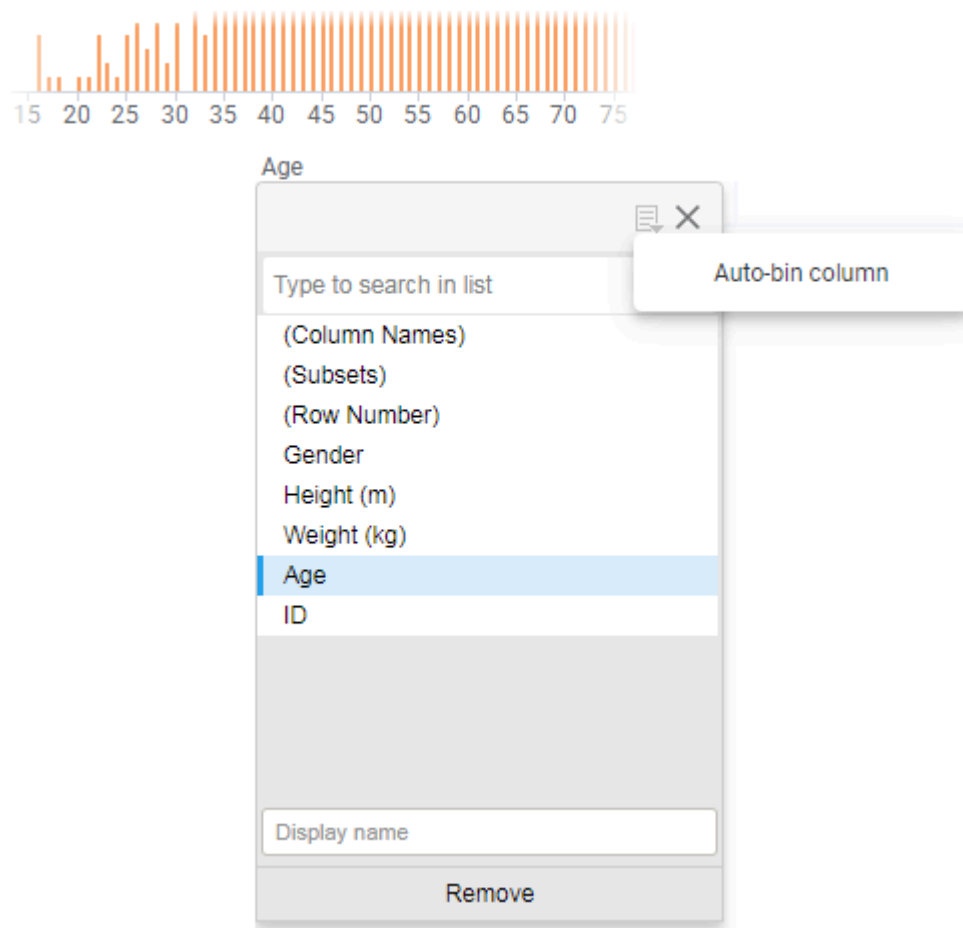
### Prerequisites

A numerical data column is selected on an axis.

### Procedure

1. Click the column selector on the axis to open its popover.

2. In the popover, click , and then select **Auto-bin column**.



The range of values on the axis is divided into bins. A slider where you can change the number of bins supplements the column selector.

3. Specify the number of bins by dragging the slider.

### Result

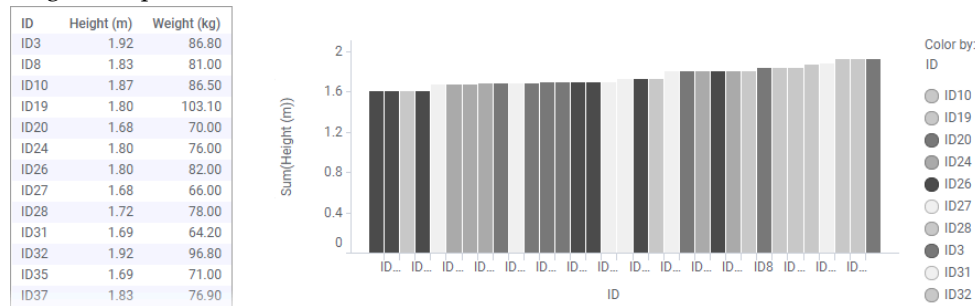
The visualization adjusts to reflect the specified number of bins. The endpoints of the intervals are automatically set to neat values.



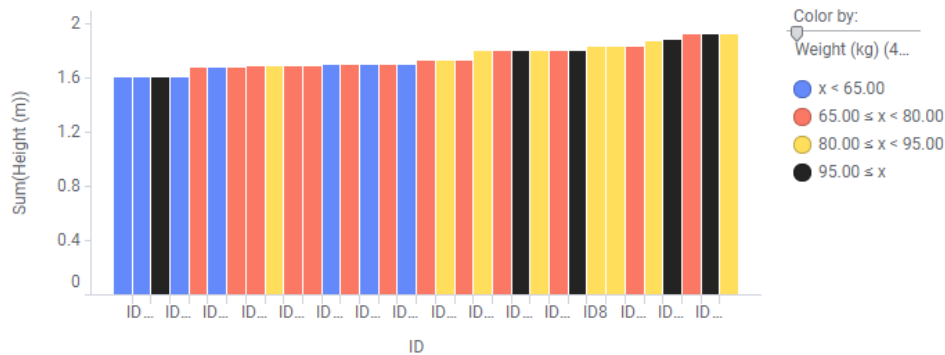
If you double-click the slider handle, you can enter the wanted number of bins in the opened dialog. Then the range between the lowest value and the highest value in the numerical data column will be divided into a number of intervals of the same size.

## Example

The table lists body heights and weights for 33 individuals, and each individual's height is represented in the sorted bar chart.



By coloring the bars by weight, and binning the numerical values, it is easy to distinguish individuals in certain weight intervals. In the image below, the range is divided into 4 bins. Individuals that weigh too much for their heights, can be spotted.



For example, the dark bar in the left part of the visualization stands out. This individual seems to weigh too much for his height.

For more examples, see [Creating a histogram](#).

## Grouping categories

Sometimes a column in your data contains many categorical values, but some of them are not important to display as separate values. Then you can group these values into a new categorical value.

For example, assume your visualization shows sales of apples, pears, oranges and limes. However, you are interested in citrus fruit sales compared to apples and pears sales. Then you can create a new category value, "citrus", that contains both oranges and limes.

When creating visualizations, you usually split data into different category values using various axis selectors. You may for example split the data by selecting a category on the X-axis, Y-axis, or the Color axis. When you find it beneficial to group category values on an axis to get fewer categories, you indicate which categories via [marking](#) items in your visualization.



In most cases marking is used to get more details about the marked items or to show relations between visualizations. In this context, the purpose of marking is to point out categories with values to group.

An example explains in more detail.

The bar chart below shows sales of toys at four different store locations. Assume you are interested in presenting the total sales at the US east coast, and therefore the east store locations should form a group (Boston and New York).

Mark these bars, right-click the visualization, and select **Group from marked categories**.



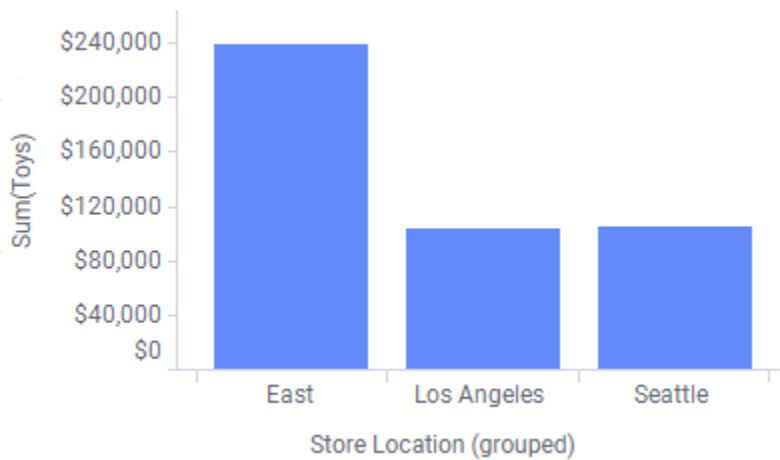
The Group from marked categories dialog opens:

Select the column from which values should be grouped. In this case the only splitting made is the Store location column on the Category axis, and the values you have pointed out by marking are "Boston" and "New York". Also give the group a suitable name.

The bar chart adjusts to reflect your grouping. The group is included in a new column, Store Location (grouped), that appears automatically on the category axis.



The column can be handled in the same way as any other column.



If you want to, you can create more groups, for example a "West" group consisting of Los Angeles and Seattle, or even a group where "East" is included.

See also [Grouping from marked categories](#) and [Handling differently spelled values](#).

## Grouping from marked categories

Sometimes a column contains many categorical values, but some of them are not important to display as separate values. Then you can group them in a new categorical value.

For more information, see [Grouping categories](#).

### Procedure

1. In the visualization, [mark items](#) you wish to group.
2. Right-click the visualization, and select **Group from marked categories**. The Group from marked categories dialog opens.
3. In the drop-down list, select the column containing the values to group.
4. Verify that the selected column is used on the correct axis in the visualization, and that the correct values are selected.
5. Give the group a name.

### Result

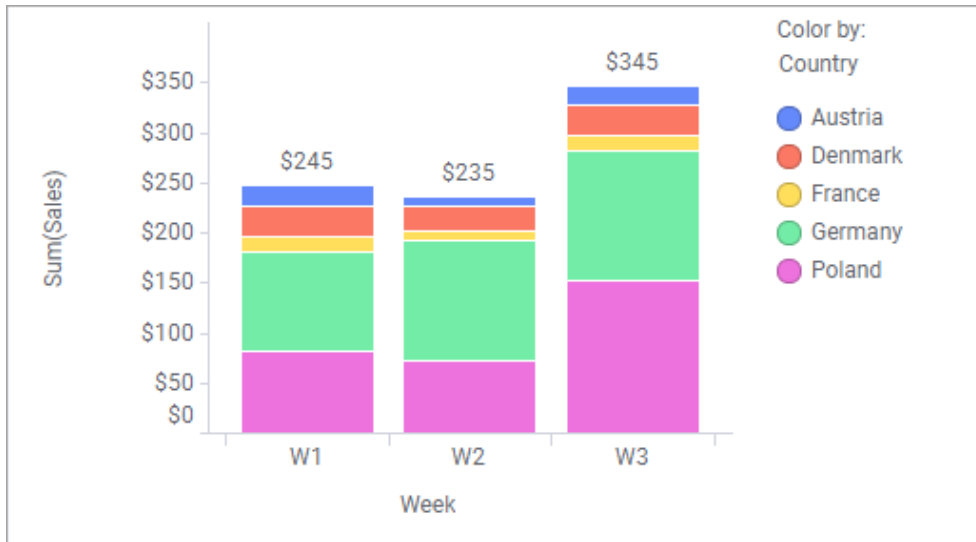
A new column containing the new group replaces the previously used column on the axis.



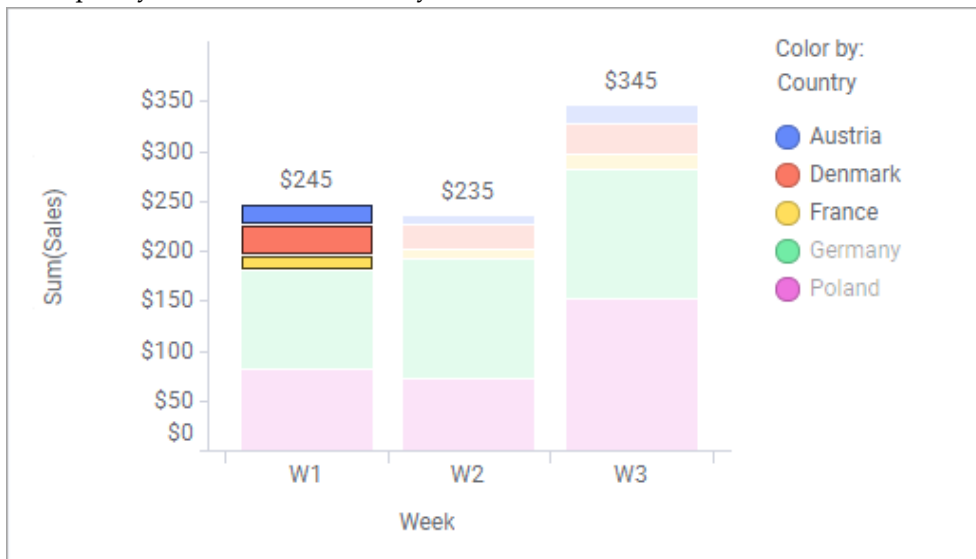
Grouping can be undone, see [Ungrouping categories](#).

### Example

The example shows first how to group categories split by the Color by axis and then how to group categories split by the category axis.



In the bar chart above, weekly sales figures are displayed, split per country. As sales figures in Austria, Denmark and France are quite low in comparison to Germany and Poland, you want to gather them in a group. To point out these countries, mark their segments. (Below, the countries are marked in the left bar, but you can mark country segments in any bar.) Note that each bar segment represents data split by both week and country.



Then right-click the visualization, and select **Group from marked categories**. Since each of the marked segments is determined by Week as well as Country, you have the option to choose from which of these columns you wish to group values. This time select Country:

Group from marked categories

Which column contains the values to group?  
Country

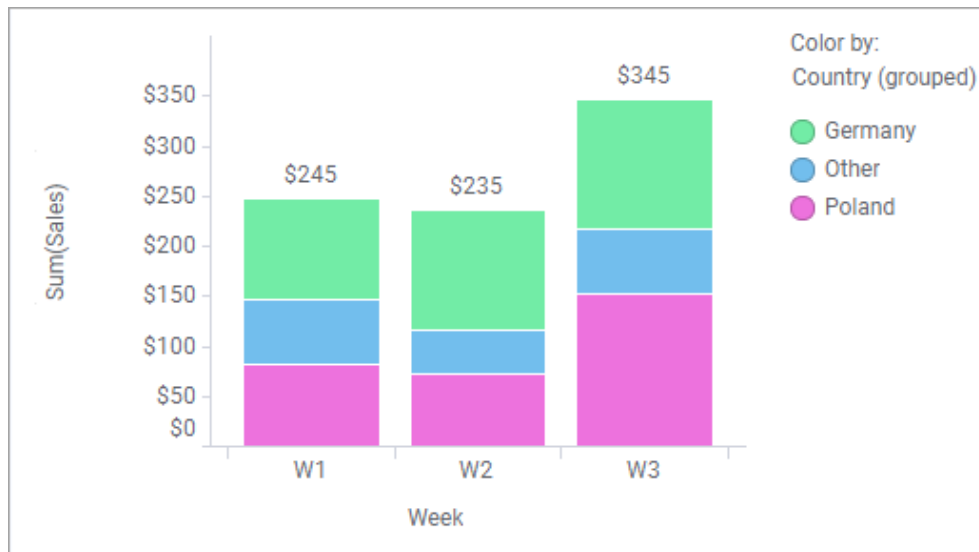
This is currently used on:  
Color by

The values you have selected:  
"France", "Denmark", "Austria"

Give the group a name:  
Other

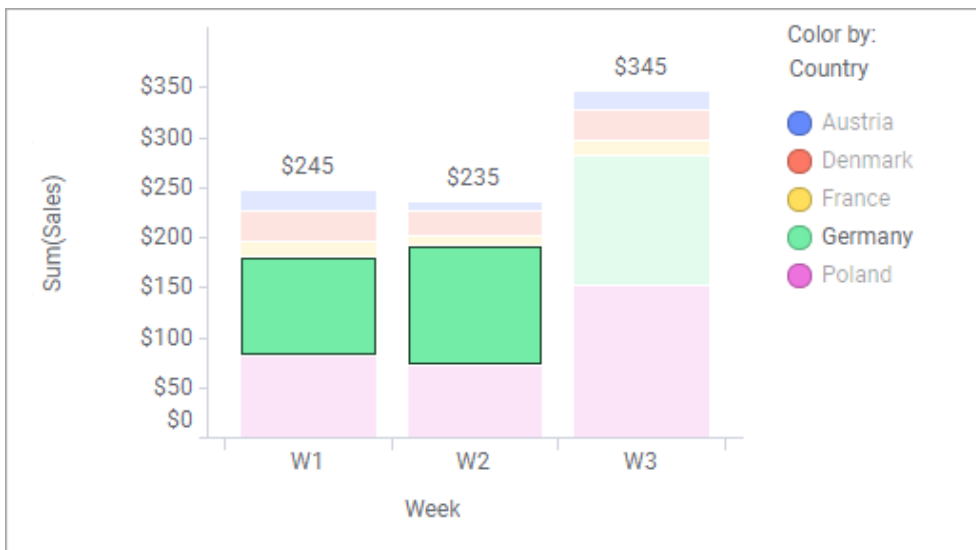
OK Cancel

The grouped countries are here named "Other". See the result with the new grouped column applied on the Color by axis:



If you wish to group weeks instead of countries, say W1 and W2, mark segments with these values. (You can mark any segment in the W1 bar, or the entire bar, and any segment/the entire bar in the W2 bar.)





Then right-click and select **Group from marked categories**.

Group from marked categories
✕

Which column contains the values to group?

Week
▼

This is currently used on:

Category axis

The values you have selected:

"W1", "W2"

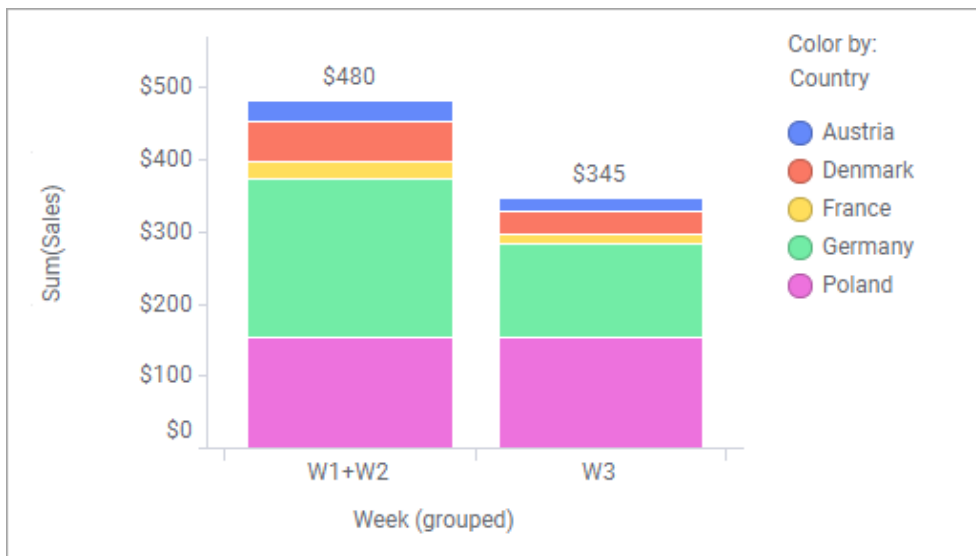
Give the group a name:

W1+W2

OK

Cancel

Here the values to group are retrieved from the Week column, and the created column grouping W1 and W2 is applied on the category axis.



## Ungrouping categories

You can undo category groupings that have been made, that is, return to the original split of the data. For more information, see [Grouping categories](#).

### Prerequisites

Values have been grouped using **Group from marked categories**.

### Procedure

1. Mark the category with grouped values.
2. Right-click the visualization, and select **Ungroup marked**.

### Result

The group is removed.

## Handling differently spelled values

If a column contains values that are spelled differently but should be identical, you can group these values in a category, and simply give the category the name you want. Values to group could be, for example, "apple" and "appel", or "UK" and "United Kingdom".

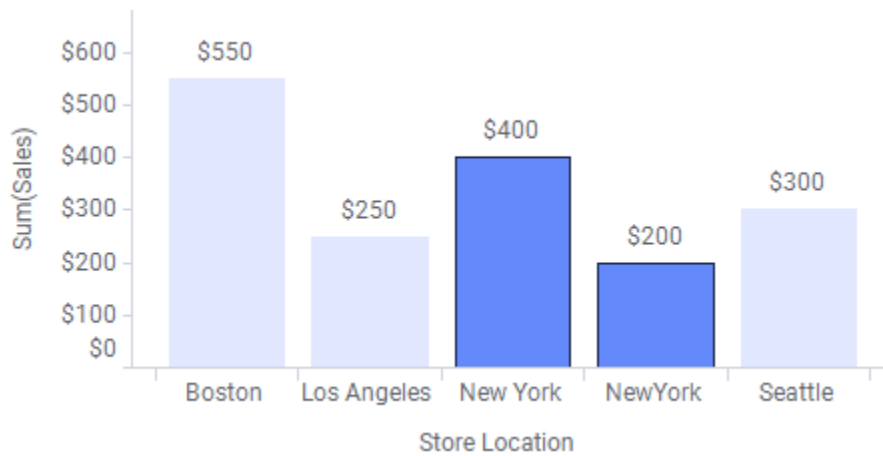
More information is found in [Grouping categories](#).



You can also handle misspelled values by [replacing the values](#) directly in a table visualization.

### Example

Below is shown how sales figures are split on different locations, but unfortunately the New York figures are split on "New York" and "NewYork".



To correct this mistake, mark these bars, right-click the visualization, and select **Group from marked categories**. In the opened dialog, give the new group containing both of the values the correct name.

Group from marked categories

Which column contains the values to group?  
Store Location

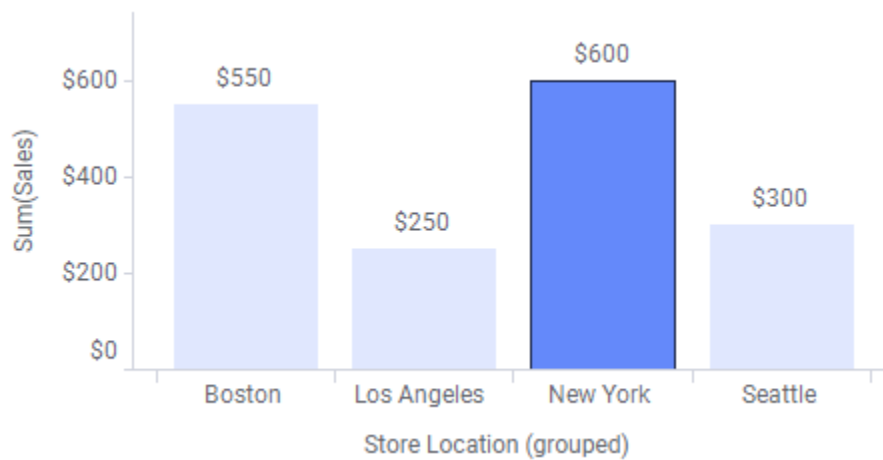
This is currently used on:  
Category axis

The values you have selected:  
"New York", "NewYork"

Give the group a name:  
New York

OK Cancel

You can see the result below.



# Making your own calculations

---

To get insight into your data, you set up visualizations using various data columns, aggregation methods, and visualization properties. Often you can retrieve even more knowledge of the data by making your own calculations on it.

When you make calculations on the data, you get the results by specifying expressions that combine one or more columns with aggregation methods and/or mathematical functions. Then you can choose different ways of exposing the results:

- [Calculated columns](#)

When you use calculated columns, the results from your expression become available as values in a new data column, which is added to the data table. The column is, like any other data column, available for selection in any visualization, and on various visualization axes, whenever the data table is used in the analysis. It can also be used in further calculations. Moreover, you can filter to the calculated values in the column, because a filter is created to represent the new column.

Generally, the calculation of the new data column is a once-only calculation. This means that once a value is calculated, it is definitive, and it will not change upon filtering. Refreshing the data leads to recalculation of the values, though.

- [Custom expressions](#)

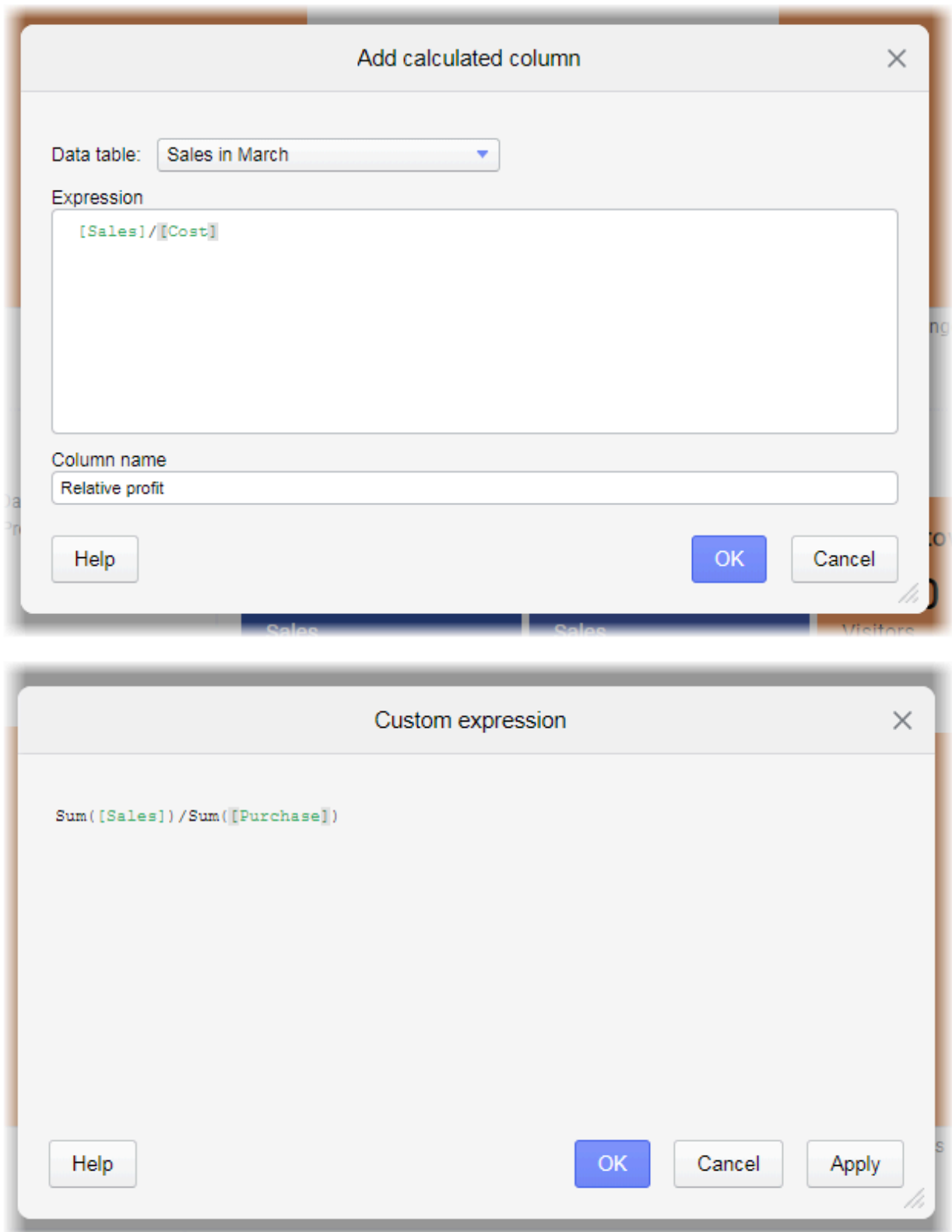
When you use custom expressions, you apply your specified expression directly to the current context, for example, to an axis in a visualization. Neither a new data column nor a filter is created, and when applied to a visualization axis, no other visualizations are affected. If the visualization properties change, for example, by filtering, your custom expression re-evaluates instantly in the same way as predefined aggregation methods. This means, you can look upon the custom expression as an aggregation method created by you.

## Common syntax

Even if these calculations are used in different ways, the expressions you create for what to calculate are common to both the methods. To get started with using expressions, you need to know some basic [syntax rules](#):

- Names of data columns must be wrapped in square brackets:
  - [Sales]
  - [Gender]
- Aggregation methods must be followed by parentheses. Parentheses enclose what to be aggregated, and they can contain more than one parameter:
  - Sum([Sales])
  - Max([Temperature])
  - Rank([Result], "desc")

Examples of the Add calculated column and Custom expression dialogs are shown below. How to access the dialogs is described in [Adding a calculated column](#) on page 730 and [Applying a custom expression](#) on page 745.



When you start typing, autocomplete functionality will suggest names of functions or column names, matching those characters you have written. Click a suggestion, and it will be added to the expression.

## Data functions

In addition to calculated columns and custom expressions, you might also have access to data functions that can provide more functionality. See [Data functions and the f\(x\) flyout](#) on page 662 for more information.

## Calculated columns

You can get more information about the data by making calculations based on the existing data columns in the data table. A calculation can contain mathematical as well as logical expressions. One way to expose the results from a calculation is to add them as values into a new data column.

The new data column will be added to the data table, and a new filter is created to represent the column. This means that you can

- select the column on the visualization axes
- include the column in further calculations
- filter to calculated values of interest.

Generally the calculation of the new data column is a once-only calculation. This means that once a value is calculated, it is definitive and will not change upon filtering. If the data is refreshed, though, a recalculation of the values takes place.

### Writing expressions

How to add the results into a new data column is described in [Adding a calculated column](#) on page 730. When you specify what to calculate, you can use various [aggregation methods](#) like sum or average to group data by, and you base the calculation on the existing data columns.

To make the expressions interpretable, you must use a certain syntax. For information about the syntax, see [Expression language details](#) on page 846. Depending on how the expression for the calculated column is set up, it is [evaluated across each row in the data table, or down an entire column](#).

If you later on want to see or edit the details of the expression used for calculating the column values, you select the column in the **Data in analysis** flyout and expand the flyout to view **Details on selected column**. See [Editing a calculated column](#) on page 731. In addition, information about the added column is available in the [Data canvas](#) on page 178.

## Adding a calculated column

Calculated columns allow you to combine or recalculate one or more columns with aggregation measures and/or mathematical expressions. This way, you can create new columns with more information based on your available data. They can be added from the menu or from the **Data canvas**.



You can also add a calculated column to a data table from the [Data canvas](#), by clicking the plus sign on the final data table node, and selecting **Add calculated column**.

### Procedure

1. On the menu bar, click **Data > Add calculated column**.
2. In the Add calculated column dialog, type the expression of interest.

For more information about which types of methods you can use, see [Functions](#) and [Operators](#).



When you start typing, autocomplete functionality will suggest names of functions or column names, matching those characters you have written. Click a suggestion, and it will be added to the expression.

3. Beneath **Column name**, give the added column a proper name.

4. Click OK.


## Editing a calculated column

When you have created a calculated column, you can edit it from within the expanded **Data in analysis** flyout, or from the **Data canvas**.



### Prerequisites

You must have at least one [calculated column](#) available and the analysis must be in **Editing** mode.



You can also edit a calculated column from the [Data canvas](#), by clicking on the final data table node, and, in the lower left part of the data canvas, clicking **Edit calculated column**  on the step representing the calculated column.

### Procedure

1. On the [authoring bar](#), click **Data in analysis**  and click on the column of interest.
2. In the expanded flyout, make sure that **Details on selected column**  is selected.

3. In the expanded flyout, locate the **Expression** section and click **Edit**.

The screenshot displays the Spotfire Web Client interface with the 'Difference' calculated column selected in the left-hand pane. The right-hand pane shows the configuration for this column. The 'Expression' section is highlighted with a red box, and the 'Edit...' button is being clicked, as indicated by a hand cursor and a tooltip that reads 'Edit calculated column'.

**Left Pane (Data in analysis...):**

- Sales Data**
- NUMBERS**
  - Cost
  - Difference** (selected)
  - Sales
- TIME**
  - Month
  - Year
- CATEGORIES**
  - Category
  - Type

**Right Pane (Configuration):**

- Difference** (with edit and delete icons)
- No description added* (with edit icon)
- Statistics:**
  - Min: -6
  - Max: 19
  - Average: 3.53
  - Median: 2.50
- Visuals:** A histogram showing the distribution of the 'Difference' values.
- Categorize column as:** Numbers
- Data type:** Integer
- Formatting:** General
- Preferred aggregation method:** Use default (Sum)
- Expression:** `[Sales] - [Cost]`
- Edit...** (highlighted with a red box and a tooltip 'Edit calculated column')
- Display values:** (Default)

**Bottom:** Clear selections



4. Change the expression and click **OK** when you are done.



For calculated columns, you cannot [change the data type](#) using a selector. Instead, use one of the [Conversion functions](#).

For more information about which types of methods you can use in expressions, see [Functions](#) and [Operators](#).

### Result

The calculated column is updated.

## Evaluations of expressions

Depending on how the expression for a calculated column is set up, it is evaluated row by row in the data table, or down an entire column. An expression can also combine these evaluations.

To illustrate how different expression setups are interpreted, the four-row data table below is used.

Fruit	Type	Purchase	Sales	Stand 1	Stand 2
Apples	Aroma	20	25	14	30
Apples	Cox orange	30	35	28	20
Pears	Anjou	15	25	10	5
Pears	Conference	25	40	22	15

The data table contains purchase and sales figures for four types of fruits. The two columns furthest to the right show the number of fruits that are sold in two different stands.

### Evaluation row by row

In the image below, two [calculated columns have been added](#) to the data table. The expressions used are based on the Sales and Purchase columns. The expression for the first column, [Sales]-[Purchase], calculates the absolute profit per fruit type, and the expression for the second column calculates the relative profit, [Sales]/[Purchase].

These expressions are evaluated row by row. They do not contain any aggregations; the individual row values in the Sales and Purchase columns are inserted in the expressions. For example, in the last row, the [Sales]-[Purchase] column shows 15 (40-25), and the [Sales]/[Purchase] column shows 1.60 (40/25).

Fruit	Type	Purchase	Sales	[Sales]-[Purchase]	[Sales]/[Purchase]
Apples	Aroma	20	25	5	1.25
Apples	Cox orange	30	35	5	1.17
Pears	Anjou	15	25	10	1.67
Pears	Conference	25	40	15	1.60

If aggregations are used in an expression, however, it is important to keep track of what data is included in the aggregation to understand what values are used in an evaluation.

In the next image, the Sum([Stand 1], [Stand 2]) and Avg([Stand 1], [Stand 2]) columns are calculated. When two or more columns are specified within the parenthesis for an aggregation method as in these expressions, then the expression is evaluated row by row.

The expression Sum([Stand 1], [Stand 2]) calculates the total number of fruits that has been sold of each type. For example, in the last row, the total sum is 37 (22+15). The other expression, Avg([Stand 1], [Stand 2]), calculates how many fruits that have been sold in average per stand for each type, such as 18.50 for Conference pears ((22+15)/2).

Fruit	Type	Stand 1	Stand 2	Sum([Stand 1], [Stand 2])	Avg([Stand 1], [Stand 2])
Apples	Aroma	14	30	44	22.00
Apples	Cox orange	28	20	48	24.00
Pears	Anjou	10	5	15	7.50
Pears	Conference	22	15	37	18.50



The expression `[Stand 1]+[Stand 2]` would give the same result as the expression `Sum([Stand 1], [Stand 2])`.

### Evaluation down an entire column

If only one column is specified within the parenthesis for an aggregation, then the expression is evaluated down the column. The expression `Sum([Sales])` in the added column below means that the values in the entire Sales column are aggregated into a sum value ( $125=25+35+25+40$ ). Then this sum value is repeated, because it is the result of the expression for every cell value in the calculated column.

The expression `Avg([Sales])` works in the same way ( $31.25=(25+35+25+40)/4$ ).

Fruit	Type	Sales	Sum([Sales])	Avg([Sales])
Apples	Aroma	25	125.00	31.25
Apples	Cox orange	35	125.00	31.25
Pears	Anjou	25	125.00	31.25
Pears	Conference	40	125.00	31.25

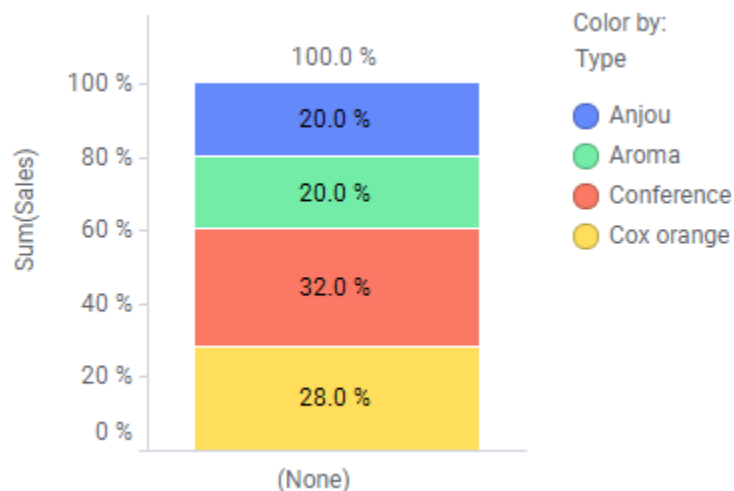
### Combination of evaluation row by row and down an entire column

In the image below, the expression `[Sales]/Sum([Sales])` for the calculated column furthest to the right combines evaluation row by row and down an entire column. The expression compares the contributions from each fruit type to the total sales on a relative basis. For example, the contribution from Conference pears to the total sales is 32% ( $40/125$ ).

Fruit	Type	Sales	Sum([Sales])	[Sales]/Sum([Sales])
Apples	Aroma	25	125.00	0.20
Apples	Cox orange	35	125.00	0.28
Pears	Anjou	25	125.00	0.20
Pears	Conference	40	125.00	0.32



Actually, this is the calculation used when setting up the 100% stacked bar chart below.



## Examples of calculated columns

You can get more information about the data by making calculations based on the existing data columns in the data table. A calculation can contain mathematical as well as logical expressions. The following examples give you an idea of how calculated columns can be used to retrieve more information about your data.

See also [Evaluations of expressions](#) on page 733.

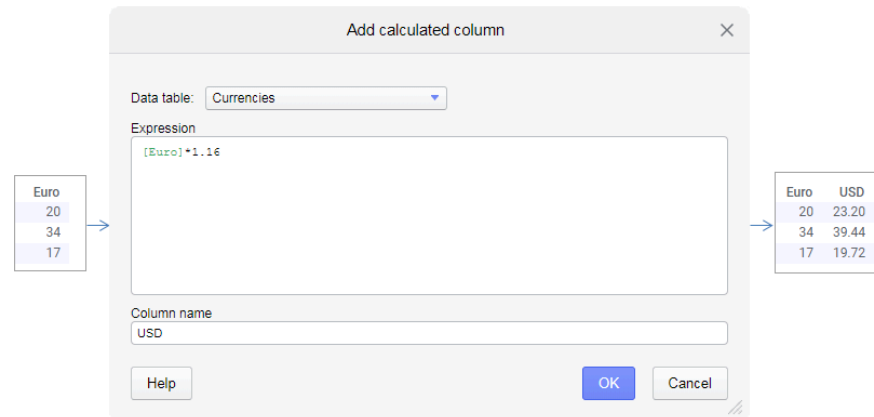
### Simple conversion and formula calculations

You might want to, for example, perform unit conversions of the values in a data column, or use formulas to calculate new values, and then add the results as values in a new data column.

The following are simple examples of retrieving more information than is available in the data table from start. New data columns, based on the existing data columns, are [calculated and added](#) to the original data table.

#### Converting currency

Below, Euro amounts in a data column are converted to USD using the exchange rate 1.16. The results are added into a new column named USD.



The calculated column is immediately available in the **Data in analysis** flyout, but it might not be visible automatically in the table visualization. To show the column in a table visualization, do the following:

1. Right-click the table, and select **Properties** in the opened menu.
2. In the popover, select **Columns**.
3. Click **Select columns**.
4. In the Select columns dialog, select the calculated column in the **Available columns** pane, and click **Add**.
5. Click **Close**.

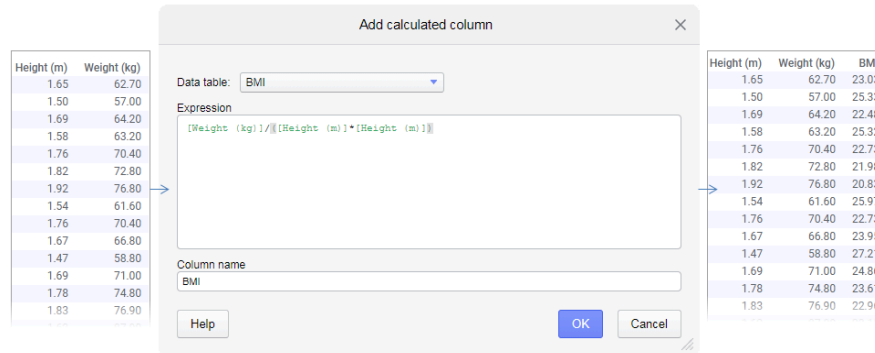


## Calculating BMI

In this example, the original data table shown to the left lists 'Height (m)' and 'Weight (kg)' for a number of individuals. Based on these data columns, Body Mass Index, BMI, can be calculated. The formula

$$[\text{Weight (kg)}] / ([\text{Height (m)}] * [\text{Height (m)}])$$

is entered in the Add calculated column dialog, and its results are added as a new data column in the data table to the right.



## Calculations of time differences

Sometimes the columns in a data table show different time points, but what is interesting is not the actual start and stop points. Instead you want to know how long different time periods lasted.

The examples below illustrate how [time functions](#) can be used to calculate time differences.

### DateDiff() function

The data table below lists start and stop times for a number of participants in a race.

Start	Stop
9:15:00 AM	11:23:00 AM
9:03:00 AM	10:09:00 AM
9:45:00 AM	11:21:00 AM
10:00:00 AM	10:58:00 AM

The actual time points are not important, but their results are. By entering the expression

```
DateDiff('Minute', [Start], [Stop])
```

in the Add calculated column dialog, the time differences between the start and stop times are calculated, expressed in minutes.

Start	Stop	DateDiff('Minute', [Start], [Stop])
9:15:00 AM	11:23:00 AM	128
9:03:00 AM	10:09:00 AM	66
9:45:00 AM	11:21:00 AM	96
10:00:00 AM	10:58:00 AM	58

### DateTimeNow() function

The `DateTimeNow()` function returns the current system time. For example, this can be used to, in combination with the `DateDiff` function described above, calculate current ages of people, if their birth dates are known.

Date of birth
6/23/1945
12/5/2004
3/29/1967
9/13/2018

By entering the expression

```
DateDiff('year', [Date of birth], DateTimeNow())
```

in the Add calculated column dialog, the time differences between current system time and the dates of birth are calculated, expressed in years.

(Current system time when the calculation below was performed was the 31st of October, 2018.)

Date of birth	DateDiff('year', [Date of birth], DateTimeNow())
6/23/1945	73.36
12/5/2004	13.90
3/29/1967	51.59
9/13/2018	0.13

### Ranking

You might want to assign a ranking number to the values within a data column.

The ranking can be made in different ways, and the examples below illustrate some of them. For more options, see [Ranking functions](#) on page 887.

## Rank() and DenseRank()

Points from a competition are listed for a number of people in the data table below. Note that two of the points are identical.

Two calculated columns are added, each of the ranking in an ascending order:

`Rank([Points])` and `DenseRank([Points])`

Name	Points	Rank([Points])	DenseRank([Points])
Charles	34	3	3
Joanna	56	5	4
Peter	34	3	3
Rosy	29	1	1
Ahmed	31	2	2

When values are identical, the two functions handle the following ranking numbers differently.

## Descending ranking

By default, the ranking is ascending. You can change to a descending order by adding 'desc' to the expressions:

`Rank([Points], 'desc')` and `DenseRank([Points], 'desc')`

Name	Points	Rank([Points], 'desc')	DenseRank([Points], 'desc')
Charles	34	2	2
Joanna	56	1	1
Peter	34	2	2
Rosy	29	5	4
Ahmed	31	4	3

### Rank groups within a column separately

Assume there are two groups represented in the competition, a and b, and they should be ranked separately.

Name	Points	Group
Charles	34	b
Joanna	56	b
Peter	34	a
Rosy	29	b
Ahmed	31	a

This can be done using the expression

```
Rank([Points], [Group])
```

shown below. To make it easier to see the separate ranking in the groups, the 'Group' column is sorted.

Name	Points	Group ▲	Rank([Points], [Group])
Peter	34	a	2
Ahmed	31	a	1
Charles	34	b	2
Joanna	56	b	3
Rosy	29	b	1

### Calculated columns based on conditions

When you calculate columns, you might want different results of an expression depending on whether certain conditions are fulfilled or not. Conditions can be set up in expressions by using the logical functions `If()` or `Case()`.

#### If()

When you use the `If()` function, you specify a condition that is either true or false, followed by what to return if the condition is true, and what to return if it is false.

## Examples

The expression

```
If([Amount]<10),"low","high")
```

means, that if a row value in the column 'Amount' is lower than 10, the value 'low' is returned as the result, but if the value is 10 or higher, the value 'high' is returned.

This is illustrated below:

Amount	If([Amount]<10),"low","high")
5	low
14	high
10	high
7	low
15	high

In other words, the `If()` function divides the rows in two groups; one group consisting of the rows for which the specified condition is true (returning 'low' as result), and the other group consisting of the rows, for which the specified condition is false (returning 'high' as result).

If the values to be returned are aggregated values, the two groups will be handled separately. This is exemplified in the table below by the added expression

```
If([Amount]<10),Avg([Amount]),Avg([Amount]))
```

that contains the average aggregations. In this case, the same aggregation is specified for both of the groups, but different aggregations can be used.

Amount	If([Amount]<10),"low","high")	If([Amount]<10),Avg([Amount]),Avg([Amount]))
5	low	6.00
14	high	13.00
10	high	13.00
7	low	6.00
15	high	13.00

For the row values less than 10, that is the 'true' group, the average 6 is returned  $((5+7)/2)$ , and for the 'false' group, the average 13  $((14+10+15)/3)$ .

The `if()` function can also be used in combination with `and` or `or`. To illustrate, the expression

```
If([Customer age]<20) and  
([Gender]="Female"),"girl","other")
```

results in the added calculated column below:





Customer age	Gender	If([Customer age]<20) and ([Gender]="Female");"girl";"other")
41	Male	other
16	Female	girl
19	Male	other
56	Female	other
19	Female	girl
21	Male	other
70	Female	other
71	Male	other
11	Female	girl

## Case()

If you want to write an expression, whose condition results are not restricted to true and false, you can use the `case()` function. `Case()` can be used in two different forms, both described in the following using examples.

### Example 1

Assume prices of books and videos below have been reduced by 10% and 50% respectively, and the price list needs an update.

Product No	Name	Type	Price
B45622	The alchemist	Book	7
A96092	The Godfather	Video	9
F987892	Harry Potter	Book	10
E456334	Harry Potter	Video	10
D867551	Da Vinci Code	Video	8
D245611	Da Vinci Code	Book	9
B005613	Superman	Comic	3

Because the reduction differs depending on the type of product, a `case()` expression is used to calculate the new prices:

```
case [Type]
when "Book" then [Price]*0.9
when "Video" then [Price]*0.5
else [Price]
end
```

Product No	Name	Type	Price	New price
B45622	The alchemist	Book	7	6.30
A96092	The Godfather	Video	9	4.50
F987892	Harry Potter	Book	10	9.00
E456334	Harry Potter	Video	10	5.00
D867551	Da Vinci Code	Video	8	4.00
D245611	Da Vinci Code	Book	9	8.10
B005613	Superman	Comic	3	3.00

Here `case()` simply checks 'Type' against each of the `when` options in the expression.

### Example 2

In the second example, `case()` is used to evaluate test results for a number of students. There are three evaluation levels; Failed (total less than 100), Passed, and Excellent (total of 160 or more).

Student	Test A	Test B
A120	67	55
A121	34	56
B786	77	93
D201	38	57
C451	93	78
D234	58	60

`Case()` in this example evaluates the conditions independently for each of the when options:

```
case
when ([Test A] + [Test B]) < 100 then "Failed"
when ([Test A] + [Test B]) >= 160 then "Excellent"
else "Passed"
end
```

This way, more complex conditions can be implemented than in example 1.

The resulting calculated column is shown below:

Student	Test A	Test B	Result
A120	67	55	Passed
A121	34	56	Failed
B786	77	93	Excellent
D201	38	57	Failed
C451	93	78	Excellent
D234	58	60	Passed



In both the `case()` forms, multiple when/then conditions can be included in the expression.

### Order execution of the conditions

Execution of the different parts of the `case()` expressions is made in top to bottom order. See the expression below, and the resulting column:

```
case
when [Number] < 40 then "yes"
when [Number] < 70 then "no"
else "x"
end
```

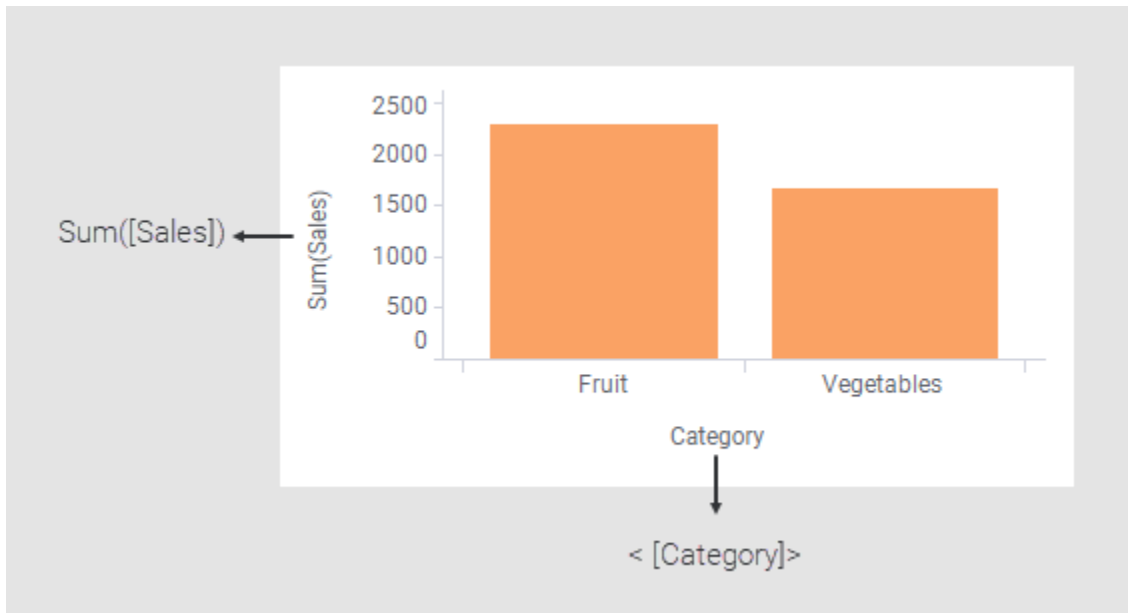
Number	Calculated column
9	yes
22	yes
34	yes
40	no
54	no
66	no
76	x
81	x

That is, numbers below 40 are set to 'yes' and will not change. The next part in the expression, set numbers below 70 to 'no', only affects the remaining rows.

## Custom expressions

You are not limited to the pre-defined aggregation methods; you can create your own expressions and apply them to the visualization axes. The custom expressions that you create can contain mathematical as well as logical parts. The result of such an expression on an axis is immediately reflected in the visualization, just as the result of any of the pre-defined aggregation methods.

Actually, as soon as you make a selection on a visualization axis, your selection automatically results in a simple custom expression that you can access. For example, if you right-click each of the column selectors in this bar chart example and select **Custom expressions**, you would see the following expressions in the Custom expressions dialogs that open:



You can anytime change the expressions on the visualization axes either by using the pre-defined aggregation methods, or by creating your own custom expressions.

Because a custom expression is applied on a certain visualization axis, only the current visualization is affected, and other visualizations in the analysis stay unaffected. The results of the expression are not added to the data table, and with that, not available elsewhere in the analysis.

Whenever the visualization properties change, for example, by filtering, re-evaluations of the custom expressions take place automatically, and only those rows of data that are available after filtering are used in the calculations.

For information on how to create a custom expression, see [Applying a custom expression](#) on page 745.

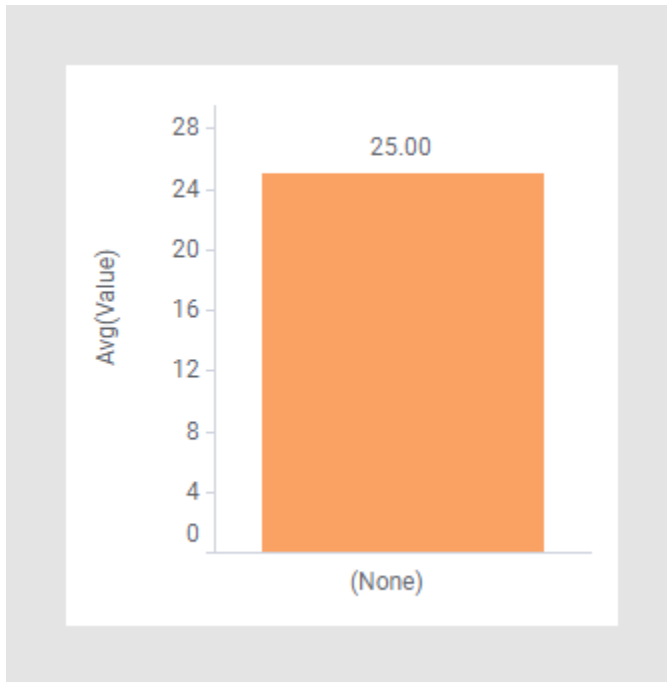
### Custom expressions work as pre-defined aggregations

Custom expressions work in the same way as predefined aggregations, with the distinction that custom expressions are not predefined by the system; they are set up by you. The similarity is illustrated in the example below that compares a well-known aggregation, average (Avg), and a custom expression that delivers the same result.

This data table contains four rows. The average value is 25,  $(10+20+30+40)/4$ .

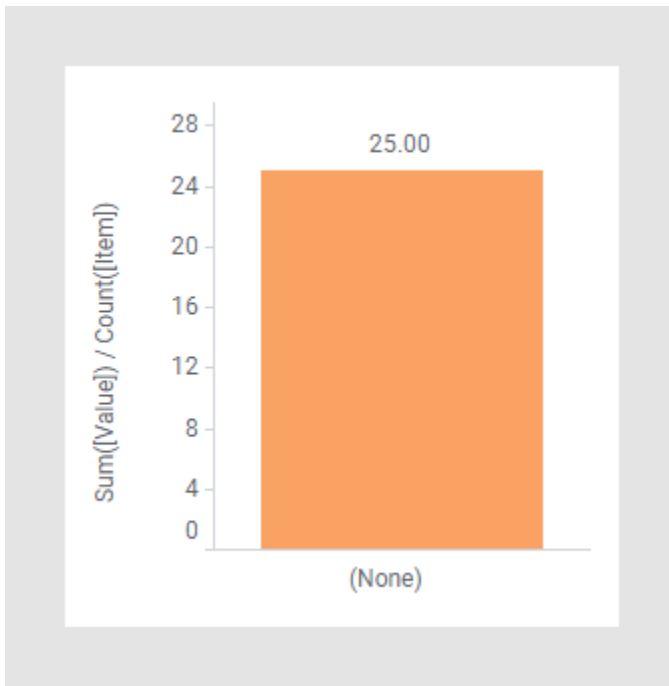
Item	Value
A	10
B	20
C	30
D	40

In the bar chart below, the result is visualized by the selected pre-defined aggregation Avg.



The next bar chart shows the same result. Instead of using the predefined aggregation, the following custom expression has been applied:

`Sum([Value])/Count([Item])`



If you filter the data, both the bar charts will adjust immediately to reflect the filtered data.

## Applying a custom expression

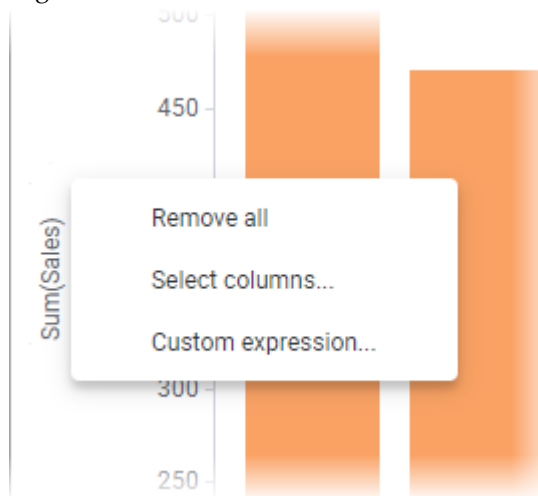
You can create your own expression using various mathematical and logical functions, and apply it directly on an axis in a visualization.



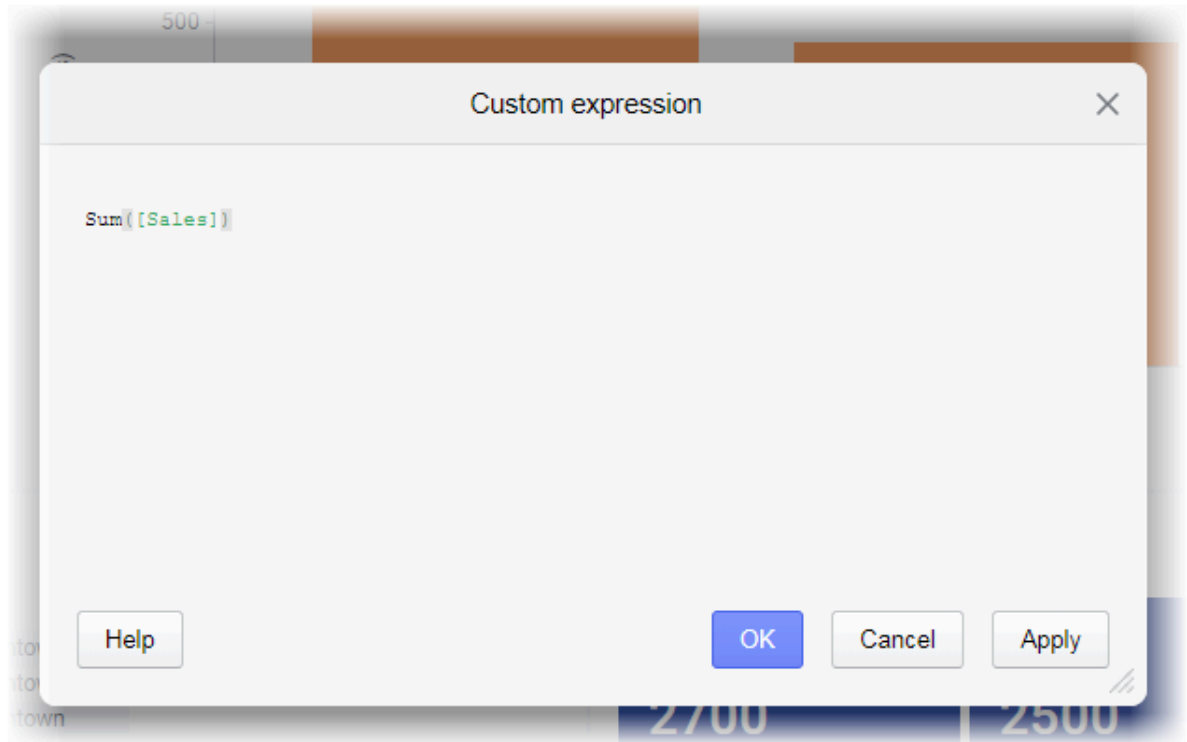
[Custom expressions](#) can be applied to axes at many different places, for example, X-axes, Y-axes, color axes, size axes, and shape axes.

### Procedure

1. Right-click the column selector for the axis where you want to use an expression of your own.



- In the menu, select **Custom expression**.  
The Custom expression dialog is displayed.



- Replace the current expression by entering your own expression. See [Making your own calculations](#) on page 728 and [Expression language details](#) on page 846.



When you start typing, autocomplete functionality will suggest names of functions or column names, matching those characters you have written. Click a suggestion, and it will be added to the expression.

- Click **OK**.  
The visualization items instantly reflect the new expression.

## Examples of custom expressions

You can create your own expressions using various mathematical and logical functions, and apply them directly on an axis in a visualization. The following examples give you an idea of how custom expressions can be used to retrieve more information about your data.

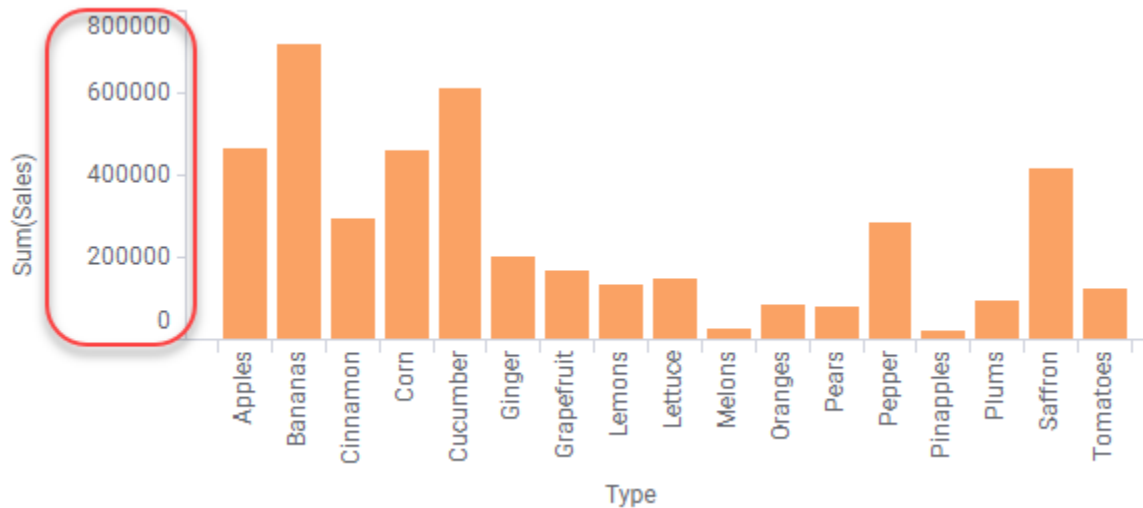
### Showing fewer digits on an axis by dividing a result with 1000

With custom expressions you can do just about anything with the columns in your data. The example below shows how you can use a custom expression to divide the result on the value axis by 1000, so that the axis value is shown in thousands of dollars instead of dollars.

The example data looks like this:

Type	Sales
Saffron	2691.88
Saffron	910.51
Plums	69.15
Plums	22.62
Apples	4141.73
Pepper	509.53
Saffron	804.86
Saffron	49.37
Saffron	1019.26
Pepper	12.60
Bananas	116.68
Oranges	128.82
Ginger	6838.33
Lemons	1161.78
Bananas	1011.56

One column contains the sales price and another the type of product.



When the sum of sales is shown on the value axis of a bar chart, a large number of unnecessary zeros are shown in the axis labels. By showing the total sum of sales in thousands of dollars instead, the numbers on the axis become smaller and easier to understand.

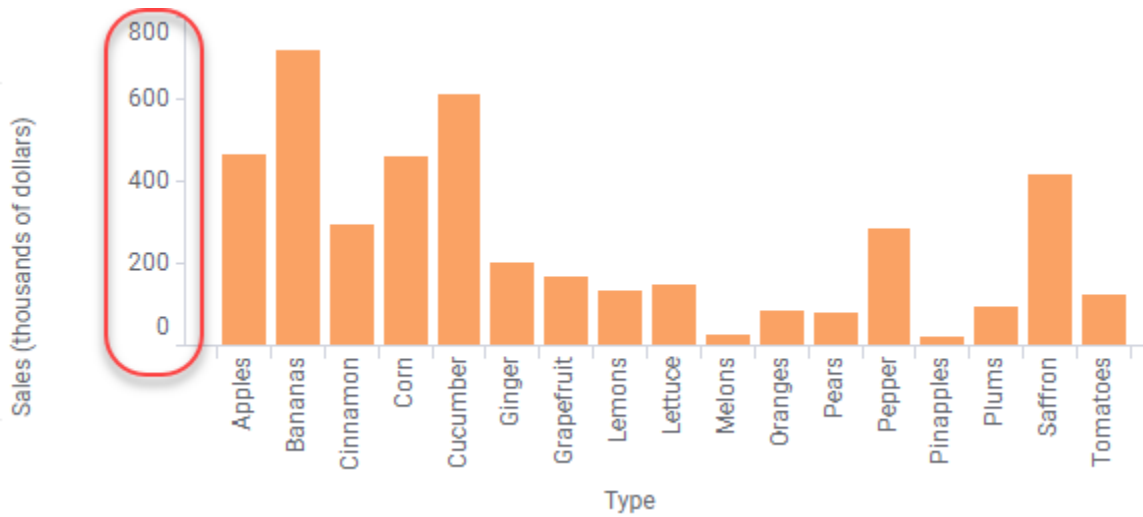
### Procedure

1. On the axis selector for the visualization where you want to divide the value, right-click to show the pop-up menu.
2. Select **Custom expression**.
3. In the Custom expression dialog, modify the expression so it says `Sum([Sales])/1000 AS [Sales (thousands of dollars)]` and click **Apply**.

The last part of this expression (AS and forward) defines what to show as the axis label and is not needed for the calculation itself.

### Result

The axis expression is changed to show sales in thousands of dollars instead of dollars.



You can also change how values are shown on an axis by [changing the formatting](#).

### Calculating the sum of two or more columns

The total sum of all values in two or more data columns can be calculated.

In the following example, the data contains Q1 and Q2 sales figures for cities in different countries. Assume you want to display the total sales for each country across the two quarters, for example, in a cross table. This means the sum of the values in the Sales Q1 and Sales Q2 columns should be totaled.

Country	City	Sales Q1	Sales Q2
Germany	Hamburg	20000	19000
Germany	Bremen	15000	21000
Norway	Bergen	40000	37000
Norway	Stavanger	35000	39000
Denmark	Esbjerg	23000	30000
Denmark	Odense	17000	25000

First, a cross table is created as shown below.

(None)	
Country	Country
	Sum(Sales Q1)
	Denmark
	Germany
	Norway
	Sum(Sales Q1)

### Procedure

1. Right-click the cell values axis to open the pop-up menu.
2. Select **Custom Expression**.
3. In the Custom Expression dialog, enter the expression `Sum([Sales Q1])+Sum([Sales Q2])`.  
The expression calculates, per country, the sum of the values in the two columns.



## 4. Click OK.

The cross table shows the total sum of Q1 and Q2 for each country.

(None)

Country	Country	Sum([Sales Q1]) + Sum([Sales Q2])
	Denmark	95000
	Germany	75000
	Norway	151000

Sum([Sales Q1]) + Sum([Sales Q2])

### Showing the difference between the values in two columns

With custom expressions you can do just about anything with the columns in your data. The example below shows how you can use custom expressions to calculate the difference between a Sales column and a Cost column to directly show the Profit for different products.

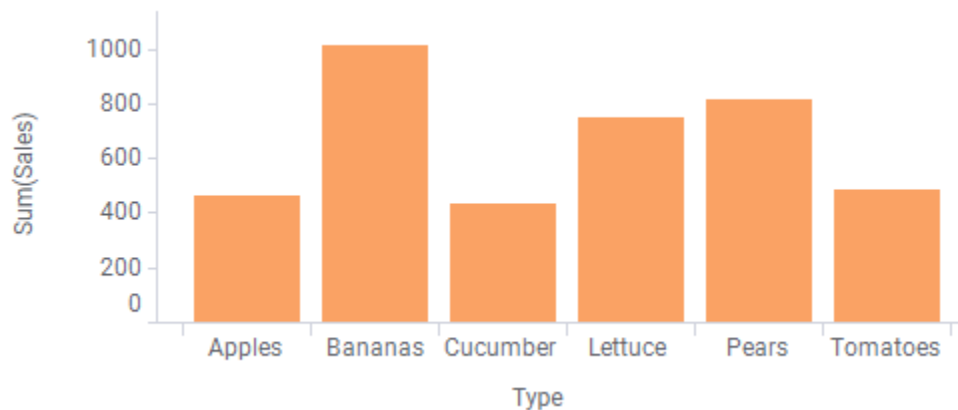


Because custom expressions work on the filtered rows, you will get different results depending on how you filter your values.

The example data looks like this:

Year	Month	Category	Type	Sales	Cost
2017	January	Fruit	Apples	13	11
2017	January	Fruit	Pears	22	14
2017	January	Fruit	Bananas	30	28
2017	January	Vegetables	Cucumber	10	7
2017	January	Vegetables	Tomatoes	14	12
2017	January	Vegetables	Lettuce	23	20
2017	February	Fruit	Apples	12	10
2017	February	Fruit	Pears	22	14
2017	February	Fruit	Bananas	32	29
2017	February	Vegetables	Cucumber	9	11
2017	February	Vegetables	Tomatoes	13	14

One column contains the sales price and another the purchase cost for each product group and month.

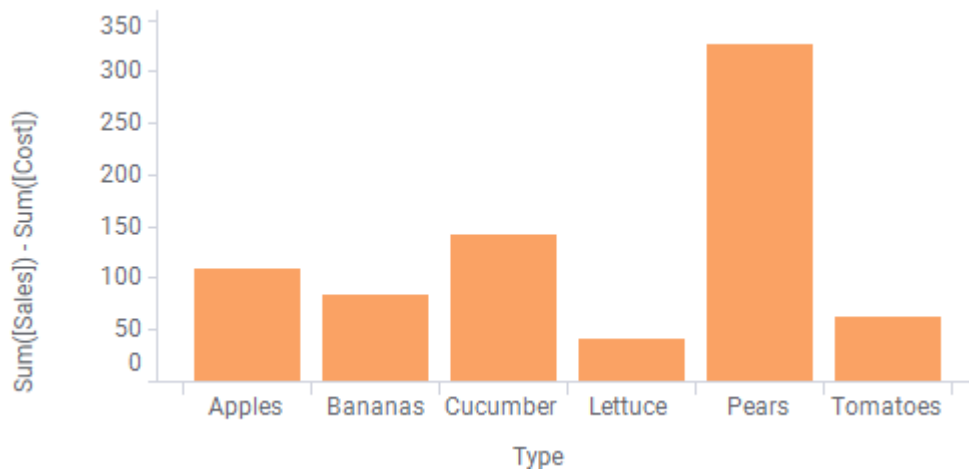


## Procedure

1. On the axis selector for the visualization where you want to show the difference, right-click to show the pop-up menu.
2. Select **Custom expression**.



3. In the Custom expression dialog, modify the expression so it says  $\text{Sum}([Sales]) - \text{Sum}([Cost])$  and click **Apply**.  
The axis expression is changed to show the difference between the values in the two columns, and it is easier to see which product earned the best profit.



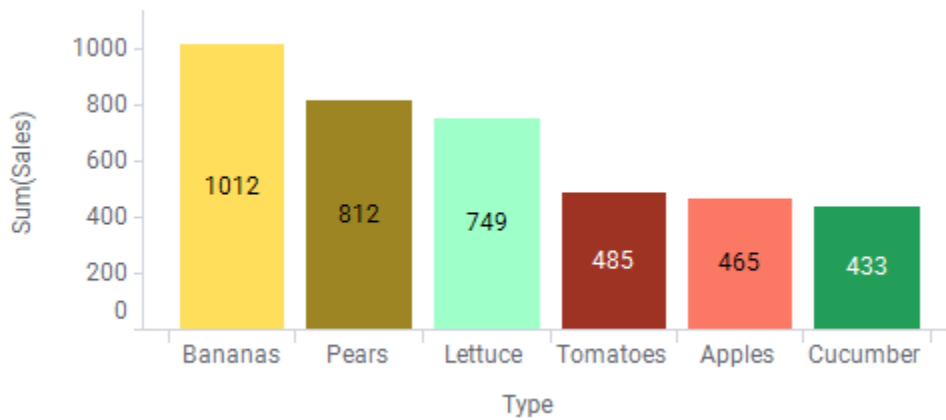
## Calculating profit in percent

The example shows how to use an custom expression to calculate the ratio between the net profit and the cost of investment.

The data table below lists sales and purchase costs for a number of fruit types.

Year	Month	Category	Type	Sales	Cost
2017	January	Fruit	Apples	13	11
2017	January	Fruit	Pears	22	14
2017	January	Fruit	Bananas	30	28
2017	January	Vegetables	Cucumber	10	7
2017	January	Vegetables	Tomatoes	14	12
2017	January	Vegetables	Lettuce	23	20
2017	February	Fruit	Apples	12	10
2017	February	Fruit	Pears	22	14
2017	February	Fruit	Bananas	32	29
2017	February	Vegetables	Cucumber	9	11
2017	February	Vegetables	Tomatoes	13	14

The bar chart, displaying the sum of sales for the different types, shows that the banana sales is highest.

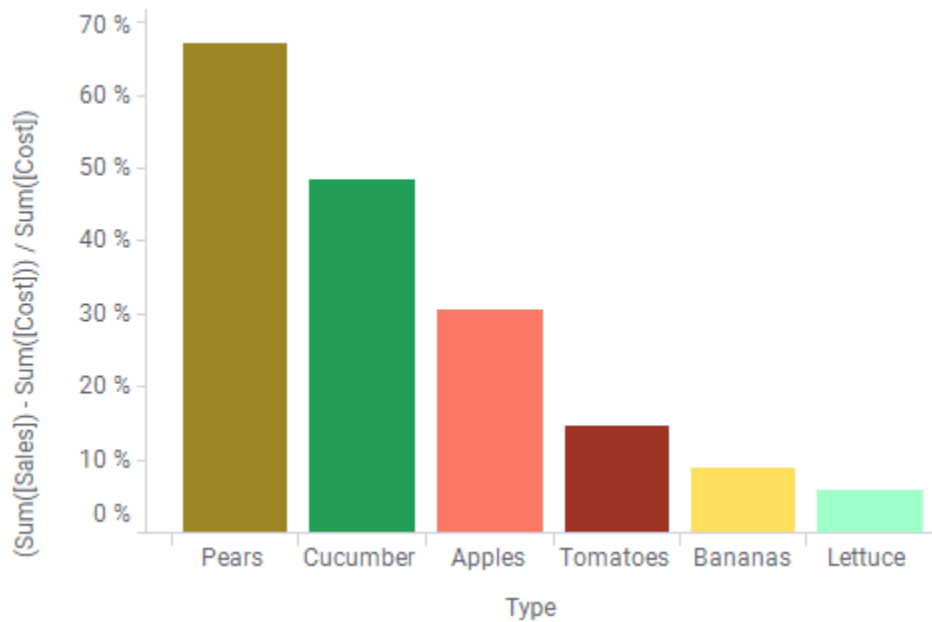


But is the bananas sales the most profitable when considering the purchase costs?

By applying the custom expression

$(\text{Sum}([\text{Sales}]) - \text{Sum}([\text{Cost}])) / \text{Sum}([\text{Cost}])$

that calculates the net profit in absolute values and then compares it to the purchase cost, the bar chart shows that sales of pears seem to be more profitable:



For every dollar you invest in pears, your net profit is \$0.67, but for bananas only \$ 0.09.

### Calculating time periods

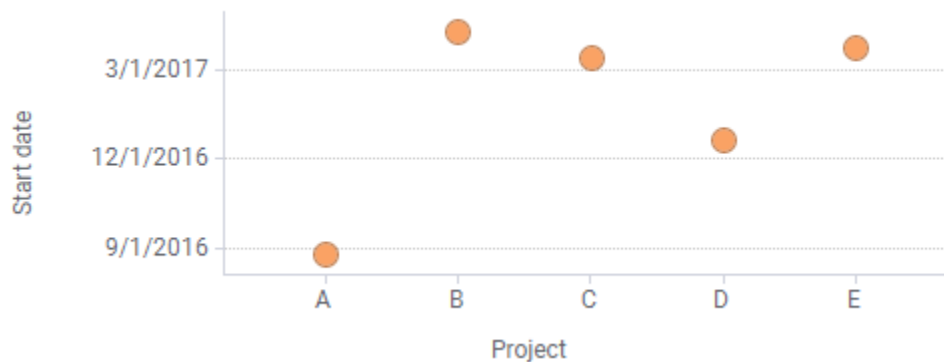
You can calculate differences between columns containing points of time to retrieve time periods. The result can be expressed in various units of time.

The function to use in the calculation is `Datediff()`.

For example, the data table below contains start and completion dates for some projects, and you might be interested in how many days each project lasted. Then, you can calculate the length of each time period between the start date and the completion date.

Project	Start date	Completion date
A	8/26/2016	8/26/2017
B	4/9/2017	4/11/2017
C	3/13/2017	4/2/2017
D	12/20/2016	2/13/2017
E	3/23/2017	5/1/2017

As a starting point, a scatter plot is created, showing one marker per project.



### Procedure

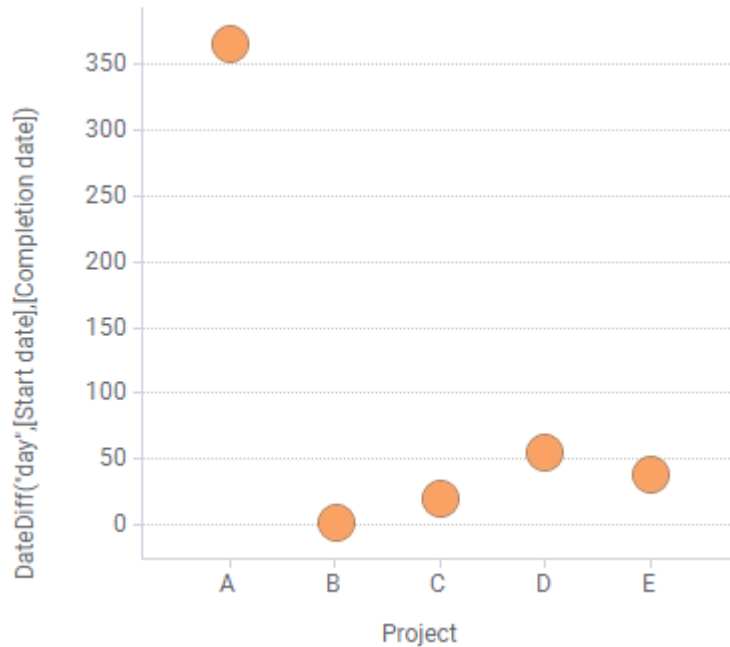
1. Right-click the Y-axis selector to open the pop-up menu.

2. Select **Custom Expression**.
3. In the Custom Expression dialog, enter the expression `DateDiff("day",[Start date],[Completion date])`.

The expression states from which date and to which date the period to calculate spans, and "day" defines which time unit to use for the result.

4. Click **OK**.

The Y-axis now shows how many days each project lasted, and it is easy to spot the project that took the longest time to complete.



## Calculating time periods between time points and today

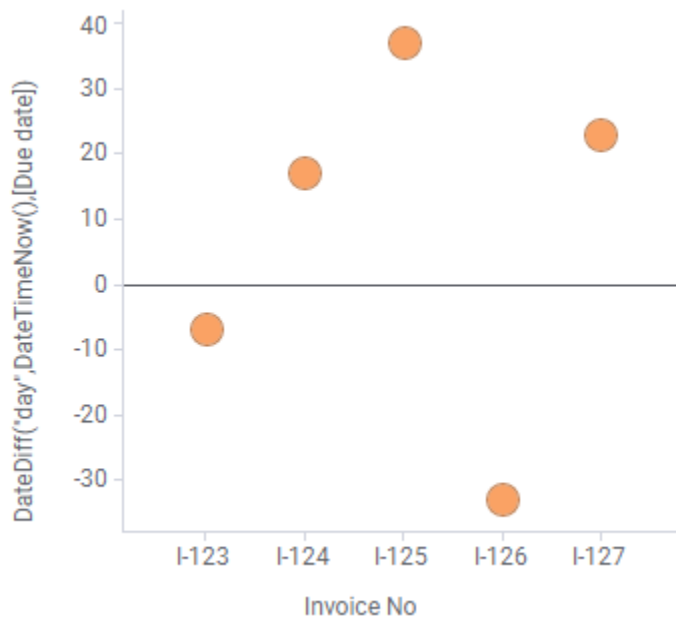
Sometimes, you are interested in the time period between a certain time point and today. For example, if a data column contains dates of birth for individuals, their current ages can be calculated. The `DateTimeNow()` function retrieves the current time in an expression.

In another example, the `DateTimeNow()` function is used to identify invoices that are urgent to pay, because their due dates are passed. The data used is shown below.

Invoice No	Due date
I-123	4/29/2018
I-124	5/23/2018
I-125	6/12/2018
I-126	4/3/2018
I-127	5/29/2018

If you apply the expression `DateDiff("day",DateTimeNow(),[Due date])`, negative differences indicate passed due dates.

The expression was used in the scatter plot below, on the 6th of May 2018, and two invoices, I-123 and I-126, with passed due dates were found.



Add a line at a certain value by clicking the axis and typing a value after **Show a line at** in the popover.

## Setting conditions on axes

When creating a visualization, you select which columns to present on the various axes, and how to aggregate the data. What is more, you can determine in more detail what to display by specifying conditional expressions on the axes. That way, what is shown in the visualizations depends on the conditions.

One function you can use for conditions in an axis expression is `If(Arg1, Arg2, Arg3)`. You interpret it as "if argument 1 is true, return argument 2 as the result, but if argument 1 is false, return argument 3".

For example, the expression `If([Amount]<10,"yes","no")` says that "if the value in the Amount column is lower than 10, the result of the expression is yes, but if the value is 10 or higher, the result is no".

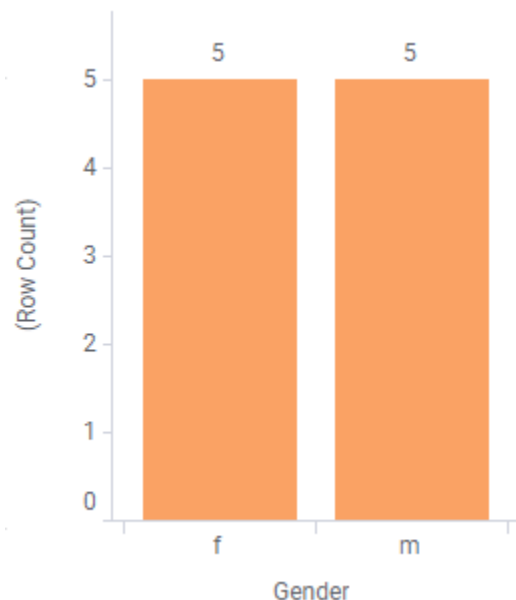
Below are some examples of how you can apply conditional expressions in different visualizations.

### Conditional expression applied on the value axis in a bar chart

The example data contains test results for a number of people. To pass the overall test, the summarized result from Test A and Test B must exceed 30. Assume you want to show how many men and how many women succeeded, and leave out those who did not pass.

Name	Test A	Test B	Gender
Paul	20	23	m
Charles	12	17	m
Lina	6	7	f
Thomas	15	22	m
Patty	11	16	f
Rosa	9	23	f
Ali	10	15	m
Tim	6	12	m
Liza	11	25	f
Alison	24	27	f

As a start, the bar chart is created with one bar per gender.



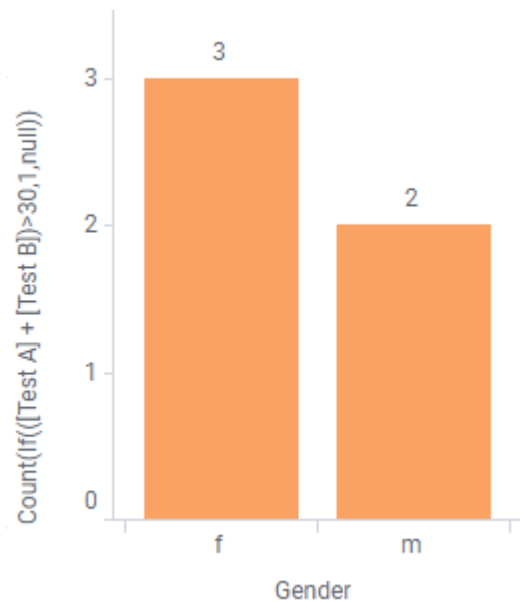
### Procedure

1. Right-click the value axis selector to open the pop-up menu.
2. Select **Custom Expression**.
3. In the Custom Expression dialog, enter the expression `Count(If(([Test A] + [Test B])>30,1,null))`.

The result of the `If` function is 1 for those who exceeded the total of 30. For those who did not pass, the Null argument indicates that no value should be returned.

Then the frequency of 1 is counted, split per gender.

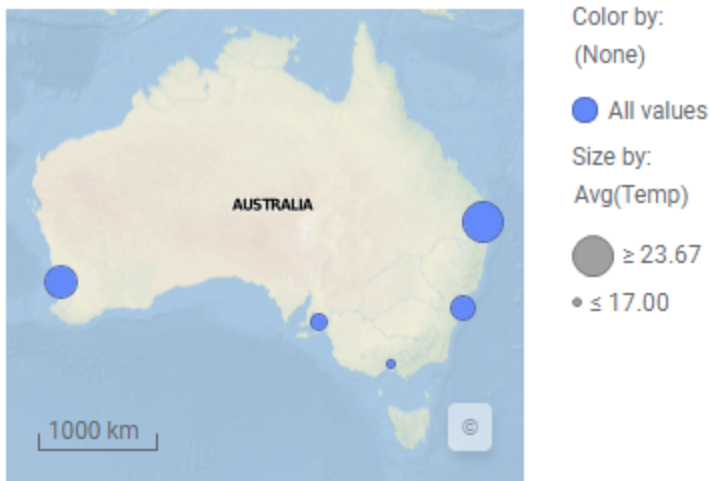
4. Click **OK**.  
The value axis expression now shows how many passed the test.



### Conditional expression applied on the color axis in a map chart

The example data contains a three-day temperature forecast for different cities. The markers in the map chart represent the average temperature for each city.

City	Date in May	Temp
Sydney	12	23
Sydney	13	20
Sydney	14	18
Adelaide	12	22
Adelaide	13	19
Adelaide	14	16
Brisbane	12	26
Brisbane	13	24
Brisbane	14	21
Melbourne	12	20
Melbourne	13	17
Melbourne	14	14
Perth	12	26
Perth	13	22
Perth	14	19



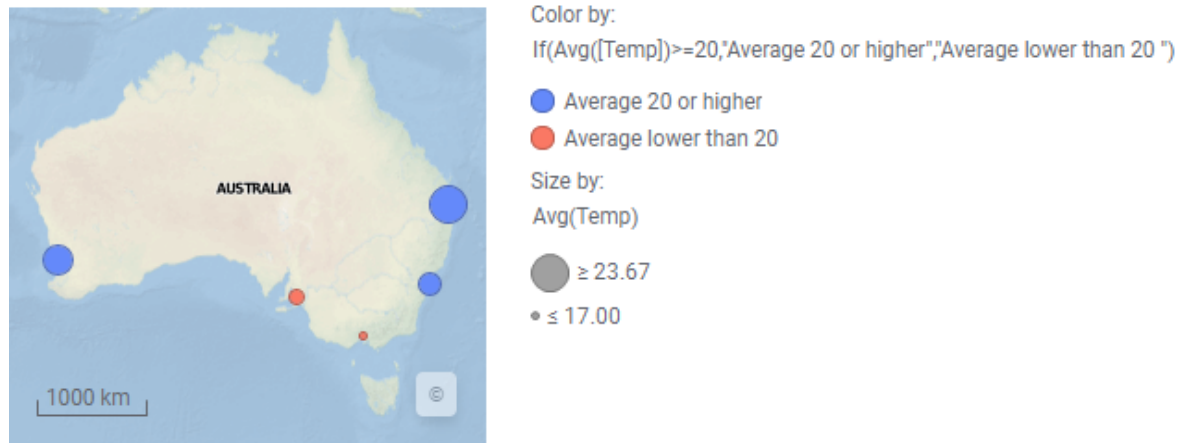
You may want to distinguish cities with an average temperature higher than 20° C by coloring their markers differently from the other markers. This can be done by applying a conditional expression on the color axis.

#### Procedure

1. Right-click the color axis selector to open the pop-up menu.
2. Select **Custom Expression**.



3. In the Custom Expression dialog, enter the expression `If (Avg([Temp]) >= 20, "Average 20 or higher", "Average lower than 20 ")`.
4. Click **OK**.  
The result of the color axis expression is a split into two categories, one category with cities fulfilling the condition, and one category that does not. The categories are colored differently.



## Slicing and marker concepts

When you visualize the data in a data table, the data is sliced into pieces, which are represented by markers in the visualization. Examples of markers are table cells, bar segments, pie sectors, map features, or line vertexes. Different marker attributes, like size, shape, color, or numerical values, are then used to indicate an aggregated value of the data underlying the marker.

The slicing and marker concepts are essential for the understanding of custom expressions, as the expressions work on the slices that have been defined in the visualization. Particularly affected are custom expressions containing [the OVER statement](#).

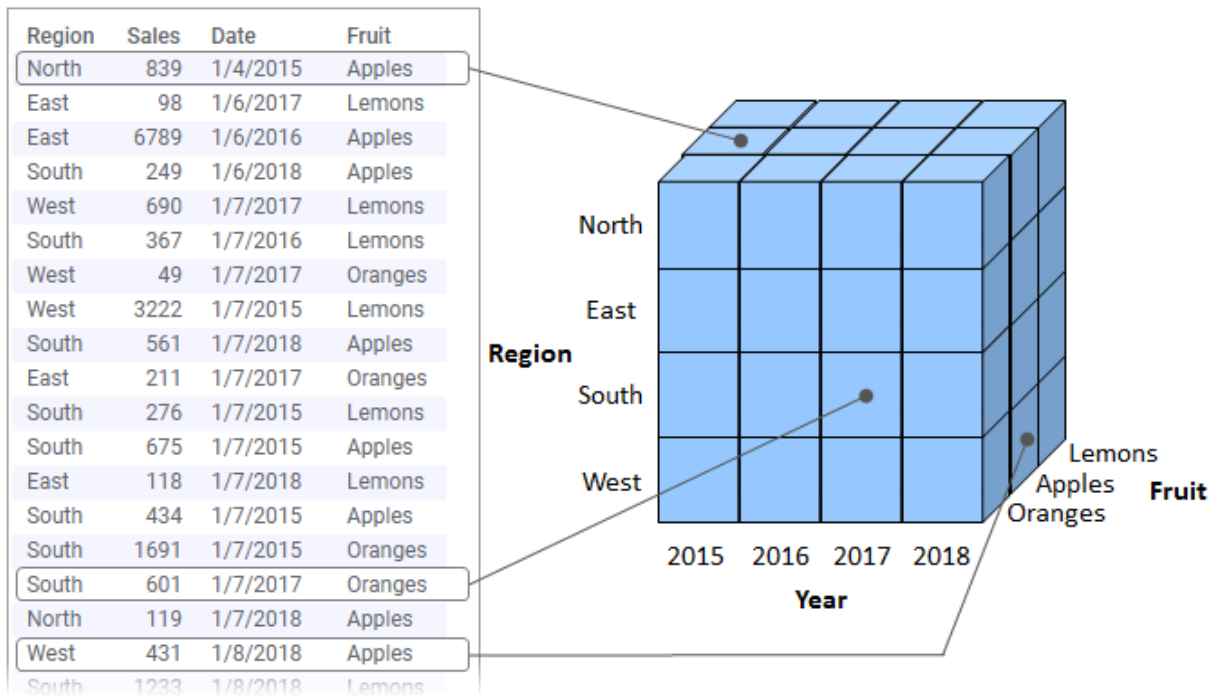
### Slicing the data

The slicing of the data into pieces is done by specifying different properties of the visualization. For example, you can slice the data on different categories on an X-axis, or slice the data by assigning different colors or shapes. What slicing properties are available depend on the type of visualization. Each time you slice the data using such a property, the data is split into smaller and smaller pieces and will be represented by a larger number of markers.

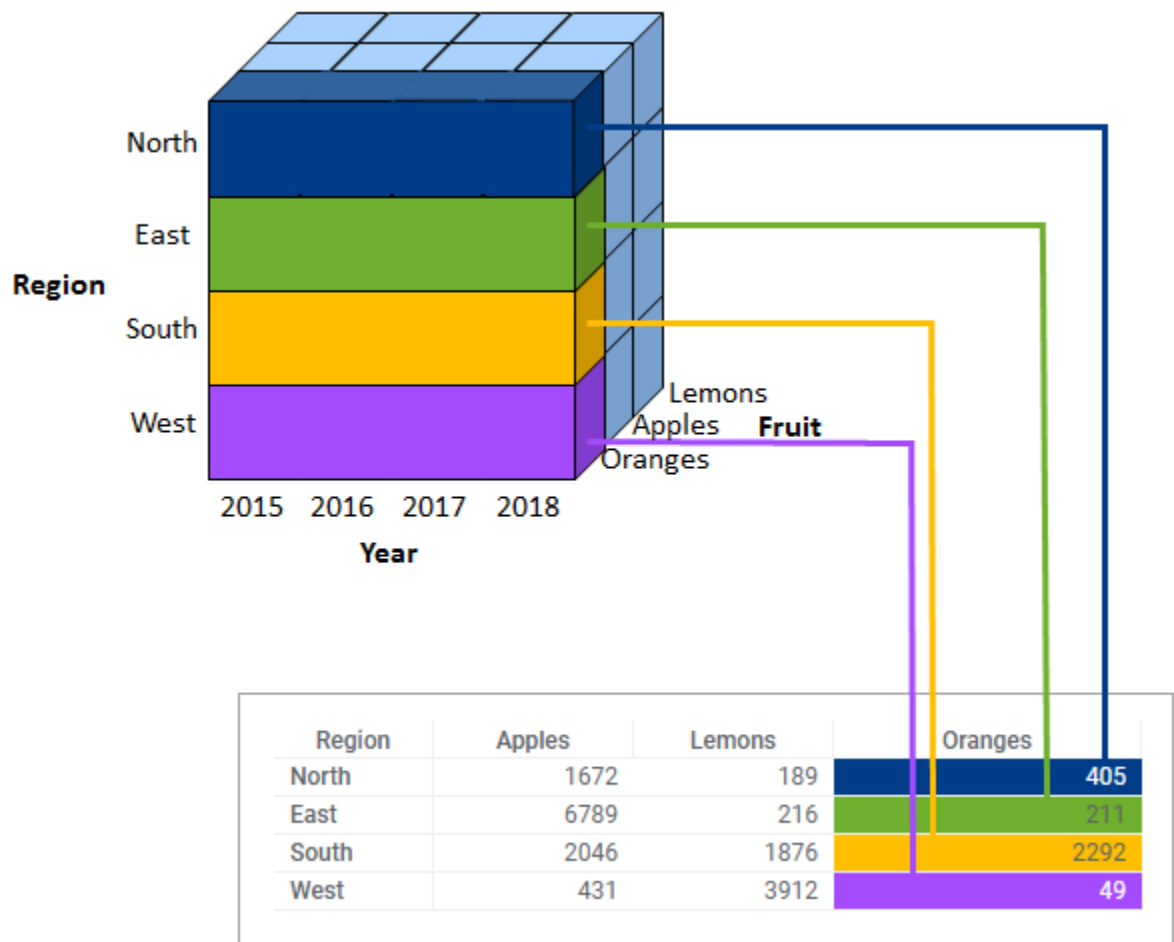
The image below gives you an idea how the data in a data table is split into pieces. The big box represents all the data, and the cubes different slices, where each of the data rows can be associated with a certain slice.



Normally, data is not restricted to three dimensions as in this example.



The following image shows how data slices can be represented in a visualization, a cross table. There is no split into years; the values in the cross table cells are aggregated over slices that comprise all the years.



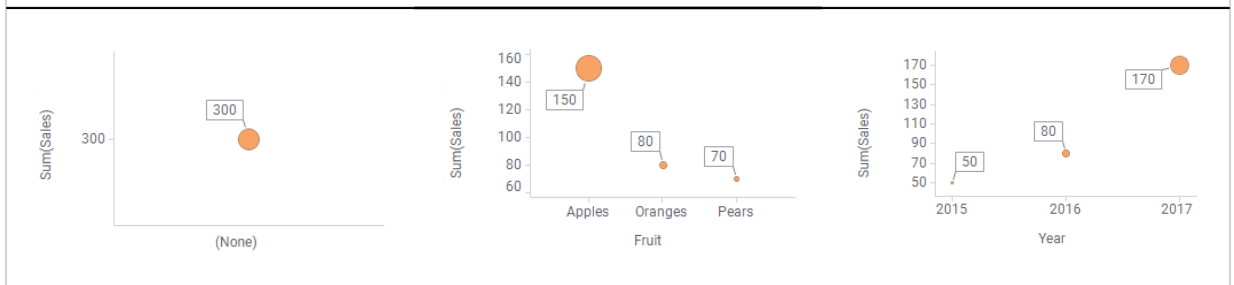
In the next example, sales data in the data table is visualized in scatter plots using different ways of slicing.

Year	Fruit	Sales
2015	Apples	30
2015	Pears	20
2016	Apples	70
2016	Pears	10
2017	Apples	50
2017	Pears	40
2017	Oranges	80

The marker represents the sum of sales of all the fruits for all years.

The markers represent the sum of sales per fruit type.

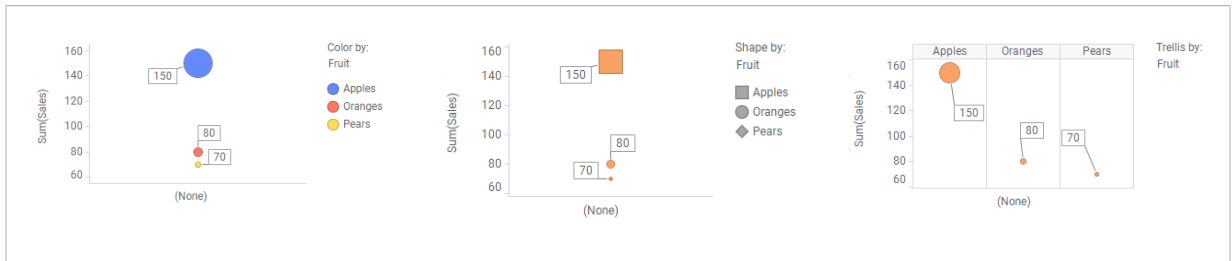
The markers represent the sum of sales per year.



In fact, your choice of slicing results in a simple custom expression: If you would right-click each of the X-axis selectors in the scatter plots, and select **Custom Expression**, you would find that the expressions `<>`, `<[Fruit]>`, and `<[Year]>` are automatically set in the respective Custom Expression dialogs.

In the same way, if you would right-click the Y-axis selector, and select **Custom Expression**, `Sum([Sales])` is shown automatically in the dialog. Aggregations like Sum are custom expressions that are pre-defined!

In the scatter plots above, the X-axis property has been used for slicing the data. However, there are also other axis properties you can use to slice by, for example, color, shape, and trellis, which is illustrated below.



To conclude, you can slice your data in various ways. The slices are then the base for your calculations.

## Markers

Markers, like the scatter plot 'symbols' and cross table cells above, are the graphical objects representing how you have sliced your data. A marker can stand for everything from all rows in your data table to a single row, depending on how you have set up your visualization, and mostly its value represents the result of an aggregation, for example, a sum as shown in the previous examples.

## Custom expressions

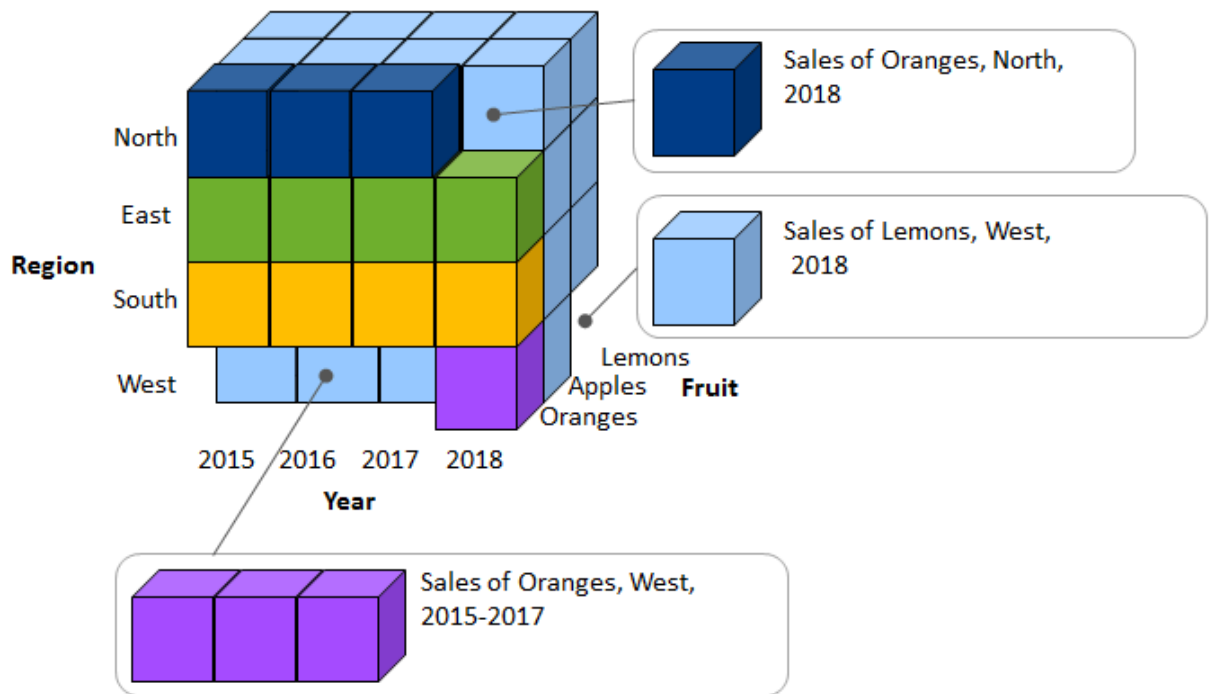
A custom expression works in the same way as a predefined aggregation; it is evaluated on each of the markers in a visualization. It is possible though to, in an expression, take in also data from outside of the current slice using [the OVER statement](#).

## Using OVER statement to reference data slices

Custom expressions are evaluated on the slices of the data, represented by the markers in the visualization. You use OVER statements in custom expressions, when you want to include also data from outside of the current slice in the evaluation of an expression.

To explain what it means to include data from outside of the current slice, see the image below. A similar image is used to illustrate [the slicing and marker concepts](#), which are essential to understand when working with custom expressions, particularly custom expressions that contain an OVER statement.

The big box symbolizes all rows in a data table containing yearly sales in different regions for some fruit types, and the cubes symbolize different slices of the data.



Assume that you want to compare Sales of Oranges, North, 2018 to Sales of Lemons, West, 2018, or to Sales of Oranges, West, 2015-2017. That is, in such an expression you need to reference data slices outside Sales of Oranges, North, 2018, which is done by using the OVER statement. The OVER statement lets you open up new ways of grouping data to be referenced in the expressions.

The easiest way to explain how to write OVER statements in expressions, and how they work, is by means of examples.

## Examples of OVER expressions



First of all, when you use the OVER statement in an expression to reference other slices, you do not refer to the actual column used for slicing the data on an axis. Instead you refer to the axis itself, like `[Axis.X]` or `[Axis.Color]`.

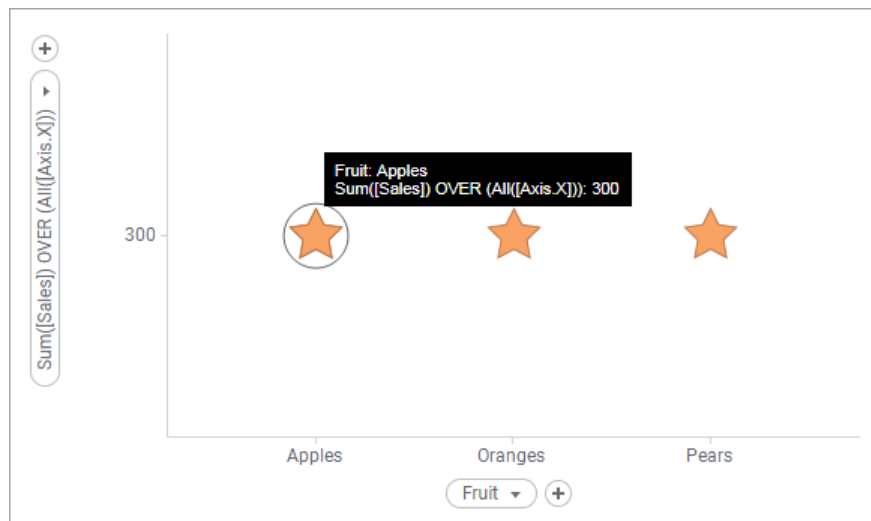
The scatter plot shows the sum of Sales for each Fruit; Apples, Oranges, and Pears. The total Sales amount is 300.



When the expression

`Sum([Sales]) OVER (All([Axis.X]))`

is specified on the Y-axis, every marker is re-evaluated, and the scatter plot changes to:



To understand the result, take the marker presenting Sum of Sales, Apples, as example. The custom expression implies that the marker shall re-evaluate to show the sum of sales for all X-axis slices, that is the sum of both Apples, Oranges, and Pears sales (150+80+70). The same goes for the other markers, so all markers end up in identical values. In fact, the OVER

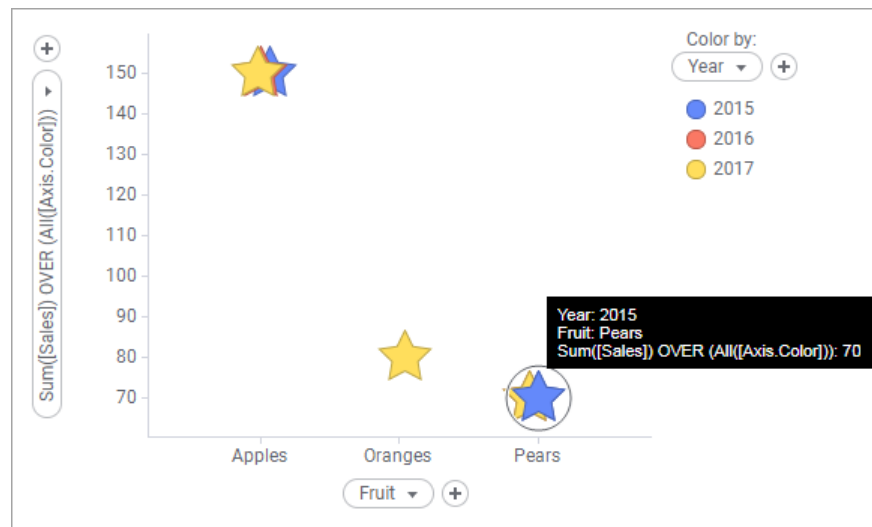
(All([Axis.X])) part of the expression implies that the slicing on the X-axis is ignored!

In the next example, further slicing of the data is made, as year is specified on the color axis.



Change the Y-axis to the following expression:

`Sum([Sales]) OVER (All([Axis.Color]))`



This custom expression implies that each marker shall re-evaluate to show the sum of sales of all color slices, as a matter of fact, ignore the slicing made by color.



The expressions used in the examples above work only when using in-memory data. If in-database data is used, a THEN expression is needed.

Take Pears, 2015, as example (the blue Pears star). The marker's new value shows the sum of pears sales for all three years (40+20+10). All markers within a fruit type get the same Y-axis value, no matter color (Year).

There is no practical use of the examples above, but they serve as a means to understand how the OVER statement works in combination with the so called node navigation methods.

## Using OVER statements in calculated columns

Using the OVER statement in calculated columns works differently from using it in custom expressions. In calculated columns, it might lead to fewer records.

### Using OVER statements in calculated columns

You can perform calculations on your data, and display the results as values in a new column that is added to the data table. In the expression used for the calculation, you can use an OVER statement. However, using the OVER statement in calculated columns works differently from using it in custom expressions. In calculated columns, the OVER expression is used to calculate separate, fixed, values for different groups and the OVER expression might cause the data to be 'sliced into fewer records'.



Calculated columns are always calculated on all values in the column, no matter how you filter your data.

When you work with calculated columns, there are no axes available to navigate over, as in the case with [custom expressions](#). Instead, you use the OVER expression to refer to a data column (or hierarchy), and you calculate one result for each category in the column or hierarchy.

Use the following data as an example:

```
Year Fruit Sales
2020 Apples 30
2020 Pears 40
2021 Apples 70
2021 Pears 10
2022 Apples 100
2022 Pears 50
```

If you create a simple calculated sum of sales, the new column will show the same result on all rows in the data table:

Year	Fruit	Sales	Sum([Sales])
2020	Apples	30	300
2020	Pears	40	300
2021	Apples	70	300
2021	Pears	10	300
2022	Apples	100	300
2022	Pears	50	300

If you instead use the expression

```
Sum([Sales]) OVER ([Fruit])
```

for the calculated column, you will get the total sum of sales for each type of fruit as the calculated result:



Year	Fruit	Sales	Sum([Sales]) OVER ([Fruit])
2020	Apples	30	200
2020	Pears	40	100
2021	Apples	70	200
2021	Pears	10	100
2022	Apples	100	200
2022	Pears	50	100

For example, as the total sales of apples is 200 (30+70+100), this value is displayed on every 'Apples' row. The effect is slicing into fewer values (200 and 100). Because you now have a column which contains an aggregated value (smearing) this means that it is probably not suitable to use such a calculated column in aggregating visualizations (for example, to show in a bar chart). See also [Using aggregations in OVER expressions](#) on page 770 for more information.



The new column values will not change when data is filtered out afterwards. For example, if the data row at the top is filtered out, the calculated column values for the 'Apples' rows will still be 200.

### Extended example

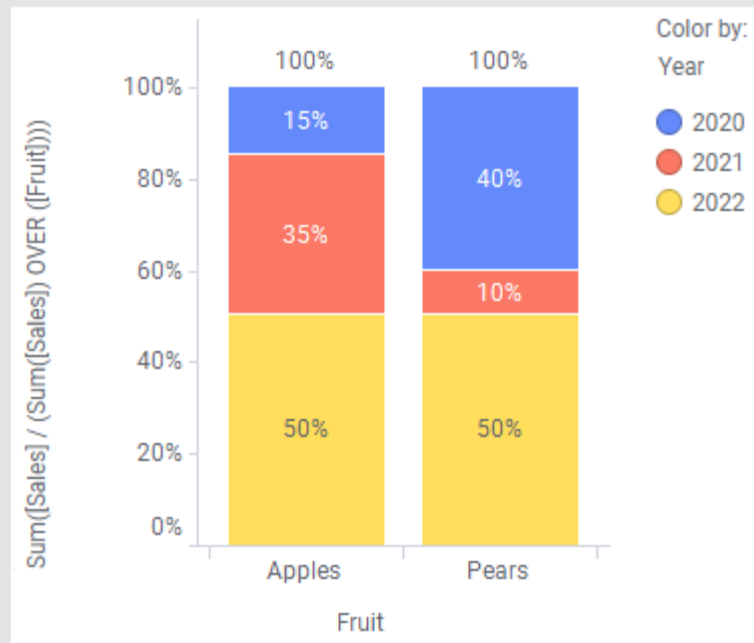
The calculated column with fewer records might not be very useful as a standalone column. However, let us add another column, where the previous expression is a part of the expression used:

`[Sales] / (Sum([Sales]) OVER ([Fruit]))`

Year	Fruit	Sales	Sum([Sales]) OVER ([Fruit])	[Sales] / (Sum([Sales]) OVER ([Fruit]))
2020	Apples	30	200	15%
2020	Pears	40	100	40%
2021	Apples	70	200	35%
2021	Pears	10	100	10%
2022	Apples	100	200	50%
2022	Pears	50	100	50%

This expression calculates each sales transaction's relative contribution to the total sales over the years within its fruit type. For example, 15% (30/200) of the total apples sales was in 2020, 35% (70/200) in 2021, and the rest, 50% (100/200), in 2022.

In the bar chart below, the calculated column is selected on the Value axis using the Sum aggregation.



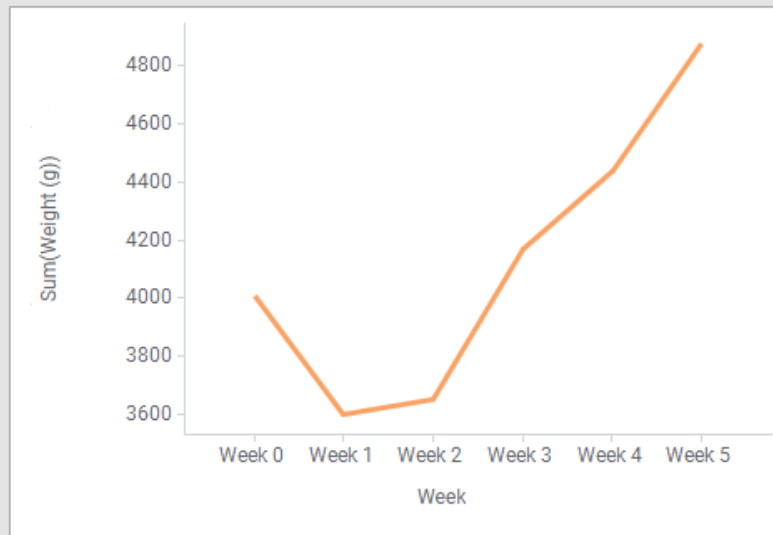
In fact, this result is the same as using the [100% stacked bars](#) option in a bar chart showing Sum([Sales]).

### Combining OVER with a node navigation method in a calculated column

In the expression used for calculating a column, you can use the OVER statement in combination with a [node navigation method](#). This simple example gives you an idea how to use one of the methods, `Previous()`.

The data table lists weekly measured weights for a newborn child, visualized in a line chart.

Week	Weight (g)
Week 0	4010
Week 1	3600
Week 2	3650
Week 3	4169
Week 4	4440
Week 5	4875

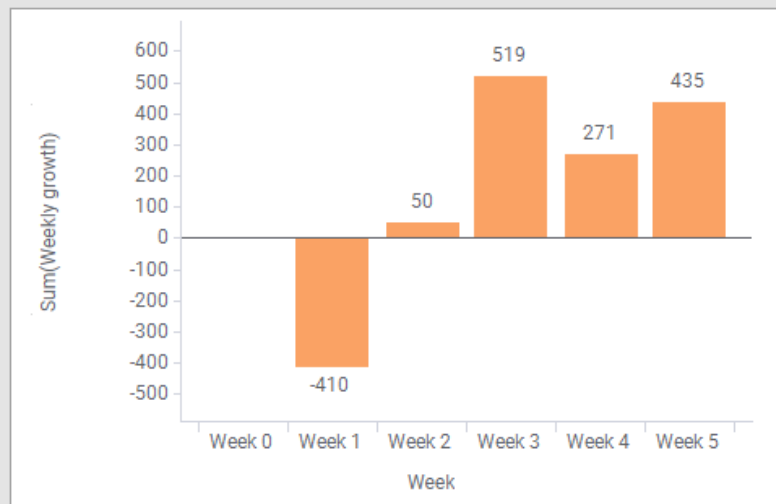


Assume you are interested in the weekly growth. This can be calculated using the expression:

```
[Weight (g)] - Sum([Weight (g)]) OVER (Previous([Week]))
```

The results can be viewed in the data table and bar chart below.

Week	Weight (g)	Weekly growth
Week 0	4010	
Week 1	3600	-410
Week 2	3650	50
Week 3	4169	519
Week 4	4440	271
Week 5	4875	435



## More examples of calculated columns

If you have multiple sales transactions each day, you could use 'OVER' to aggregate those values into one value for each day, the daily sales, using the expression `Sum([Sales]) OVER [Date]`:

Date	Sales	Sum([Sales]) Over [Date]
10/26/2017	10	10.00
10/27/2017	15	25.00
10/27/2017	10	25.00
10/28/2017	25	25.00

By combining OVER with functions such as `All`, `AllPrevious`, `LastPeriods`, etc., you can obtain just about any type of grouped values. For example, the expression `Sum([Sales]) OVER AllPrevious([Date])` will give you the cumulative sum of sales for all dates in the column 'Date'.

Date	Sales	Sum([Sales]) OVER AllPrevious([Date])
10/26/2017	10	10.00
10/27/2017	15	35.00
10/27/2017	10	35.00
10/28/2017	25	60.00

If you want subgroups for the results to be defined by values in more than one column, and the values from either of those columns make use of any of the OVER functions, then you must also use the `Intersect` function. For example, to be able to obtain the cumulative oil production per well and year in a data table with oil production data for a number of wells, you would need to intersect the well and year:

```
Sum([Oil Production]) OVER Intersect([Well ID],
AllPrevious([Year])).
```

When you are referring to a hierarchy within an expression, the nodes in that hierarchy become similar to the nodes on an axis (compare to custom expressions), and you can specify which nodes to compare using functions such as `All`, `Previous`, `Parent`, etc. Hierarchies are referenced in expressions using the syntax `Hierarchy.HierarchyName`. For example, with a hierarchy called 'TimeHierarchy', containing Year and Quarter, the expression `Sum([Sales]) OVER (Parent([Hierarchy.TimeHierarchy]))` would return the sum of sales per year.

Example	Expression
Average sales per category and year	<code>Avg([Sales]) OVER ([Category], [Year])</code>

Example	Expression
Average sales per category for all years, up until the current year	<code>Avg([Sales]) OVER Intersect([Category], AllPrevious([Year]))</code>
Peak oil per well	<code>Max([Oil Production]) OVER ([Well ID])</code>
Cumulative oil production per well and year	<code>Sum([Oil Production]) OVER Intersect([Well ID], AllPrevious([Year]))</code>
Sum of sales for top level of a hierarchy	<code>Sum([Sales]) OVER (Parent([Hierarchy.TimeHierarchy]))</code>

## Using aggregations in OVER expressions

When the results of an expression should be an aggregated view of the values in a column, you simply use the expression function of your choice to determine the output. For example, when a column containing sales figures should be summarized, you enter `Sum([Sales])` in the expression field. However, aggregated values combined with OVER expressions allow you to split the data the way you want it.

### Nested aggregations

There are a few differences in how nested aggregations behave when used in calculated columns and in custom expressions. See below for some examples. (Copy the text below and paste it (installed client only) or save it as a text file to use as a data table in an analysis and try it out yourself.)

```
Region Product Sales
EAST Phone 42
EAST Phone 118
EAST Chair 190
WEST Desk 19
WEST Chair 20
CENTRAL Phone 18
CENTRAL Desk 17
CENTRAL Desk 12
CENTRAL Desk 13
CENTRAL Chair 20
```

### In calculated columns: Max of total Sales per Region

By writing function calls within functions you can get complete control over the output of your expression. More than one aggregation can be nested, and each nested aggregation can have its own OVER expression. When the data table is filtered, the calculated column still shows values based on the whole data table.

See also [Using OVER statements in calculated columns](#) on page 764.



Calculated columns based on in-database data do not support OVER expressions or nested aggregations. However, both constructions work fine when used in the context of a visualization, using custom expressions.

Below is an example of how you can use the Add calculated column tool to compute the total sales amount per region, and then find the max of these sales.

## Procedure

1. Load the data above in an analysis.
2. Select **Data > Add calculated column** on the menu bar.
3. In the **Expression** field, type:

```
Max(Sum([Sales]) OVER ([Region]))
```

4. Click **OK**.

## Result

The new column is added to the data table. You can now use the maximum value in comparisons in different visualizations.

### Calculation details

This is what happens within the data engine when a calculated column is added; note that the intermediate tables shown in these examples are never presented to end users.

Original data:

Region	Product	Sales
EAST	Phone	42
EAST	Phone	118
EAST	Chair	190
WEST	Desk	19
WEST	Chair	20
CENTRAL	Phone	18
CENTRAL	Desk	17
CENTRAL	Desk	12
CENTRAL	Desk	13
CENTRAL	Chair	20

First the inner aggregation will be computed, and the sum of sales per region will be:

Region	Total sales (TEMPORARY)
EAST	350.00
WEST	39.00
CENTRAL	80.00

In a calculated column, the result from a calculation is always applied to all rows of the input table, hence, the resulting sales value for each region will be smeared over all the rows:



Region	Product	Sales	Total sales (SMEARED)
EAST	Phone	42	350
EAST	Phone	118	350
EAST	Chair	190	350
WEST	Desk	19	39
WEST	Chair	20	39
CENTRAL	Phone	18	80
CENTRAL	Desk	17	80
CENTRAL	Desk	12	80
CENTRAL	Desk	13	80
CENTRAL	Chair	20	80

Max ( ) will then be computed on "Total Sales (SMEARED)", resulting in the value 350, and because we are inserting a calculated column, that value will also be smeared over all rows. This means that the final result will be:

Region	Product	Sales	MAX(Sum([Sales]) OVER ([Region]))
EAST	Phone	42	350
EAST	Phone	118	350
EAST	Chair	190	350
WEST	Desk	19	350
WEST	Chair	20	350
CENTRAL	Phone	18	350
CENTRAL	Desk	17	350
CENTRAL	Desk	12	350
CENTRAL	Desk	13	350
CENTRAL	Chair	20	350

When the data table is filtered, the calculated column still shows values based on the whole data table.

### In custom expressions: Sales for the best selling Product per Region and filter out one product group

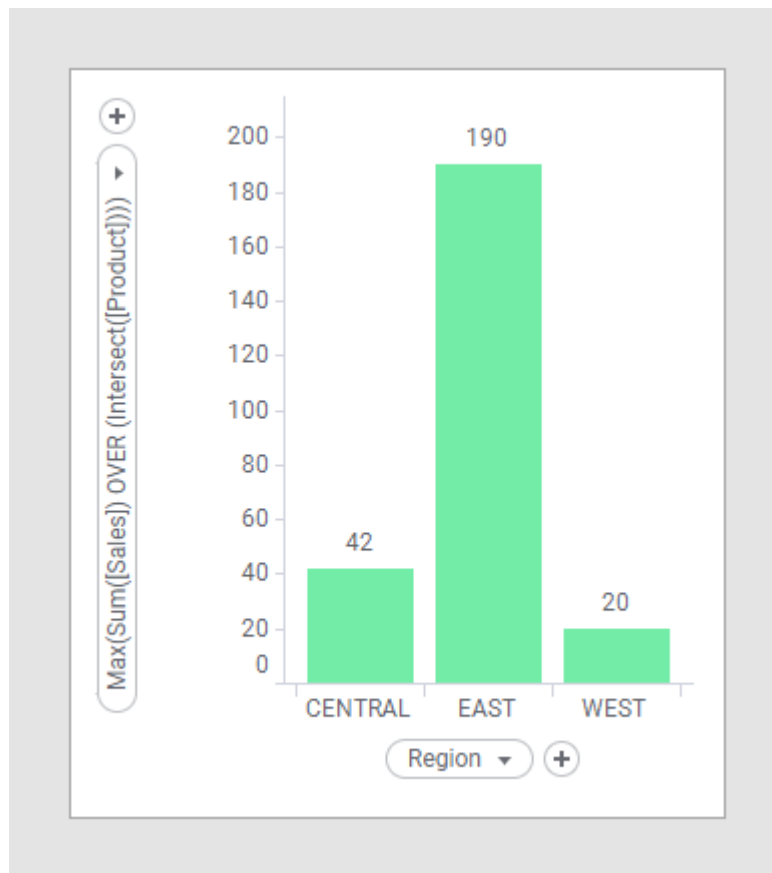
Nested aggregations can also be used in custom expressions on axes in visualizations. The behavior is similar to the case with calculated columns, but there are some differences. First of all, filtering is taken into account, so all aggregations will work on the filtered subset of the data. Also, only the outermost aggregation will use the grouping defined by the visualization, while all the inner aggregations need OVER expressions to split the data. All inner aggregations will be treated the same way as calculated columns, meaning that resulting values of the inner aggregation will be added to all rows of the outer aggregation. When there is an OVER expression in an inner aggregation, all the categorical axes in the visualization will automatically be intersected with (not for unaggregated visualizations), even though this is not explicitly stated in the expression. See *Calculation details* below for more information.

In this example, we want to show one bar per region in a visualization, where the height of the bar is the sales for the best selling product in that region. We also want to filter out the product "Chair", assuming that it is the top product besides chairs that is of interest.

### Procedure

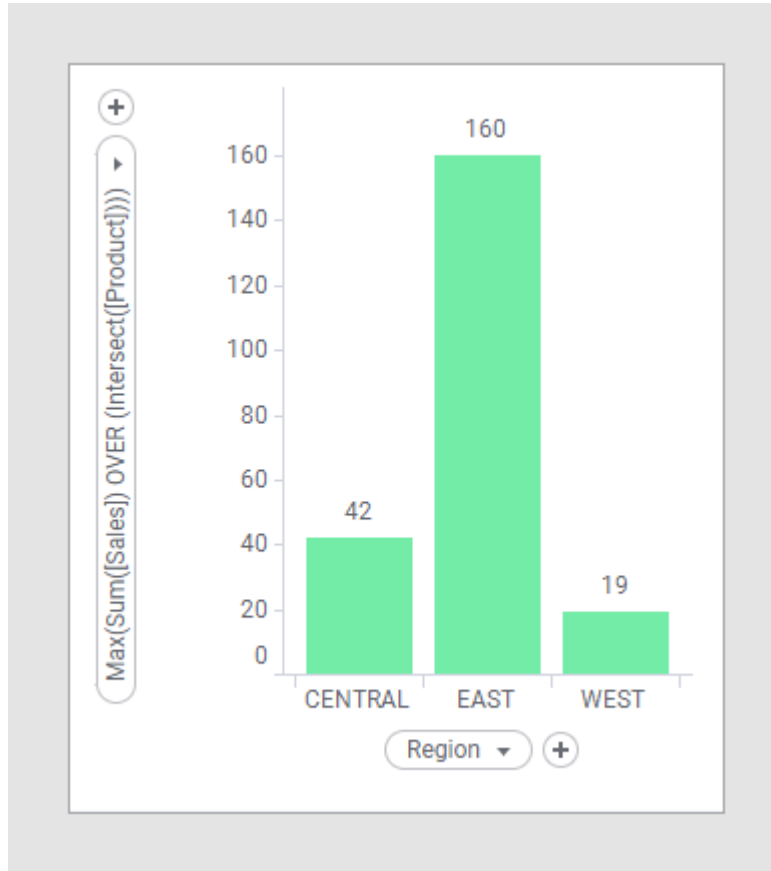
1. Load the data above in an analysis.
2. Create a bar chart.
3. On the category axis, use Region.
4. On the value axis, use the following expression to show the max sum per product:

```
Max(Sum([Sales]) OVER Intersect([Product]))
```



The bar chart is showing the values for the best-selling product in each region.

5. In the **Filters** panel, clear the Chair check box.



### Result

The bar heights are recalculated based on the filtered data table and a new max value is shown.

### Calculation details

This is what happens within the data engine when a calculated column is added; note that the intermediate tables are never presented to end users.

Starting with the original data, you have the following table:

Region	Product	Sales
EAST	Phone	42
EAST	Phone	118
EAST	Chair	190
WEST	Desk	19
WEST	Chair	20
CENTRAL	Phone	18
CENTRAL	Desk	17
CENTRAL	Desk	12
CENTRAL	Desk	13
CENTRAL	Chair	20

When "Chair" is filtered out, the data on which the calculations are based in the visualization is:

Region	Product	Sales
EAST	Phone	42
EAST	Phone	118
WEST	Desk	19
CENTRAL	Phone	18
CENTRAL	Desk	17
CENTRAL	Desk	12
CENTRAL	Desk	13

The inner aggregation will be intersected with the categorical axis, hence, the actual expression computed will be: `Sum([Sales]) OVER Intersect([Region],[Product])`, which will result in the following sales per (region,product)-pair:

Region	Product	Sum([Sales]) OVER Intersect([Region],[Product])
EAST	Phone	160
WEST	Desk	19
CENTRAL	Phone	18
CENTRAL	Desk	42

Just as for calculated columns, inner aggregations will always be smeared to all rows before the result is passed to the outer aggregation. This means that the following will be the input to the `Max()` function:

Region	Product	Sum([Sales]) OVER Intersect([Region],[Product])
EAST	Phone	160
EAST	Phone	160
WEST	Desk	19
CENTRAL	Phone	18
CENTRAL	Desk	42
CENTRAL	Desk	42
CENTRAL	Desk	42

Since this example is calculated as a custom expression in a visualization, the outermost aggregation (`Max()`) will use the grouping defined by the visualization. This means that the actual expression calculated will be based on the filtered, smeared rows, which will result in the following values for the bars:

EAST	160
WEST	19
CENTRAL	42

Because the "Chair" category was filtered out, the top remaining category in the east region was "Phone" and the top category in the west region was "Desk". If the filtering is changed, the resulting top category changes as well.

When applicable (depending on your data and which axis you are looking at), there is an expression shortcut called **Top Category** available that uses nested aggregations to help you find the top category. It has been designed to handle multiple top categories and allows you to easily change the inner aggregation as well as the category to compare. It uses the following expression:

```
UniqueConcatenate(If(Max(Sum([Sales])
OVER (Intersect([Product]))) OVER
(Intersect())=Sum([Sales]) OVER
(Intersect([Product])),[Product],null))
```

### Using expressions on aggregated data (the THEN keyword)

The use of the THEN keyword makes it possible to evaluate an expression on already aggregated data, usually the data in the current visualization. In many cases, this can give better performance in the calculations. For example, when using OVER expressions there is always a gain to be done by first allowing the data to be aggregated and then applying the OVER part of the expression. This is required

when working with in-database data because there is no row-level data available in that case, but performance can be improved when working with in-memory data as well.

In a THEN expression, the already aggregated data is referred to as a temporary column called [Value]. That is, [Value] represents the result of the entire expression before the THEN keyword.

The part of the expression after the THEN uses the result of the part of the expression before THEN, as if it was a column. For example, a custom expression calculating a cumulative sum of sales would be:

```
Sum([Sales]) THEN Sum([Value] OVER (AllPrevious([Axis.X])))
```

This expression works for both in-memory and in-database data, whereas the similar `Sum([Sales]) OVER (AllPrevious([Axis.X]))` only works in memory.

In the THEN expression above, `Sum([Sales])` is evaluated first and aggregates data into the temporary column [Value]. The expression following THEN is then calculated based on this temporary column.

## Syntax

- Aggregated expression (THEN expression) AS [Identifier]
- [Value] can be used to refer to the value of the previous expression (either the main expression or a previous THEN expression; there can be multiple THEN in the same expression).
- Values on specific hierarchy levels can be referred to using the syntax [X. LevelName] where this represents the value of level "LevelName" in the axis "X" hierarchy. You can use one level reference for each level in each axis hierarchy. For example, to refer to the year in a date hierarchy, use [X. Year].
- In THEN expressions you might sometimes need to pick a value from a specific node. In this case you can use [CategoryIndex.X] to pick out the node of interest in an expression. For example, to get the value from node number five on an axis and put it on all other nodes you could use an expression like this:

```
Sum([Sales]) THEN Sum(if([CategoryIndex.X]=4,[Value],0)) OVER (All([Axis.X]))
```

The category index for picking the fifth node is set to four, because the first node on the axis will get the category index 0.

THEN expressions can be used on all continuous axes and expressions, like labels, rules etc. However, they cannot be used on categorical axes or together with error bars.



In the installed client, if your current expression results in numerical values, but the visualization treats them like categories, you can often right-click on the axis in the visualization and select **Continuous Scale**.



Only use OVER in post-aggregate expressions, if possible.

## Examples

Instead of: `Sum([Sales]) OVER (AllPrevious([Axis.X]))`

Use: `Sum([Sales]) THEN Sum([Value]) OVER (AllPrevious([Axis.X]))`

To produce a cumulative unique count for string columns you could use something like this:

```
UniqueCount([Column]) THEN Sum([Value]) OVER (AllPrevious([Axis.X])) as [Cumulative Count]
```



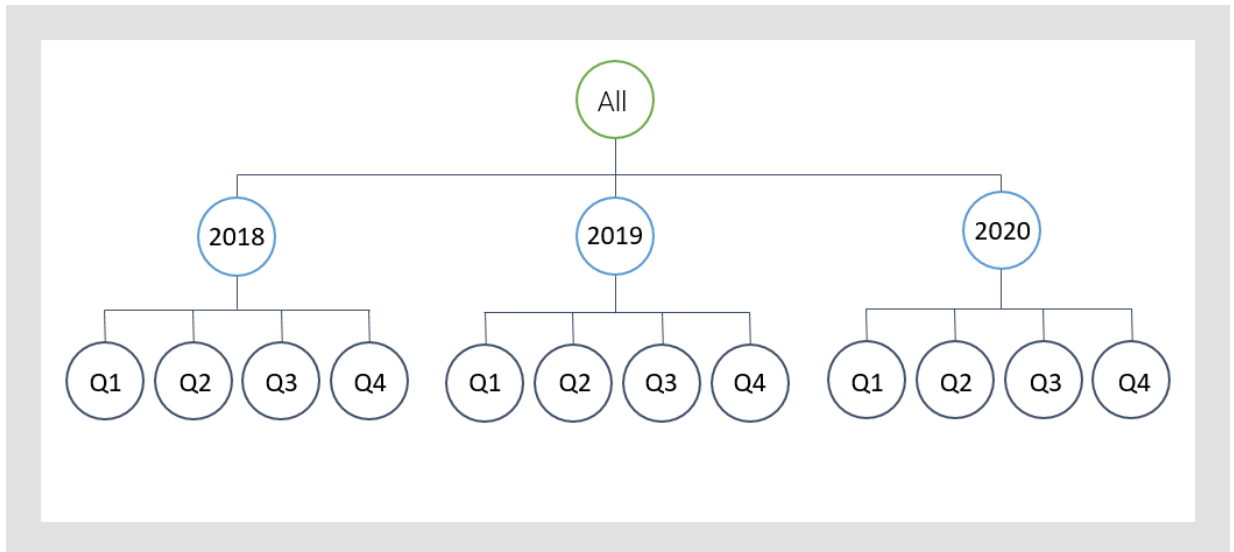
Some axis expressions might need to be evaluated after filtering to produce the correct results. In the installed client, you can right-click on an axis and select **Evaluate Axis On > Current Filtering Only** to change the axis evaluation setting in a visualization.

See also [Shortcuts to frequent custom expressions](#) on page 808 for more information about expressions using THEN.

## Node navigation

When you visualize data, it is sliced into pieces, which are represented by different types of markers, for example, table cells, bar segments, line vertexes, or pie sectors. When you want to perform calculations that involve various combinations of slices, there are different methods available to reference these combinations. The methods are called node navigation methods, and they are used together with the OVER statement.

The following illustration will serve as a base to clarify the different node navigation methods. It shows a time hierarchy where nodes represent various slices of the data.

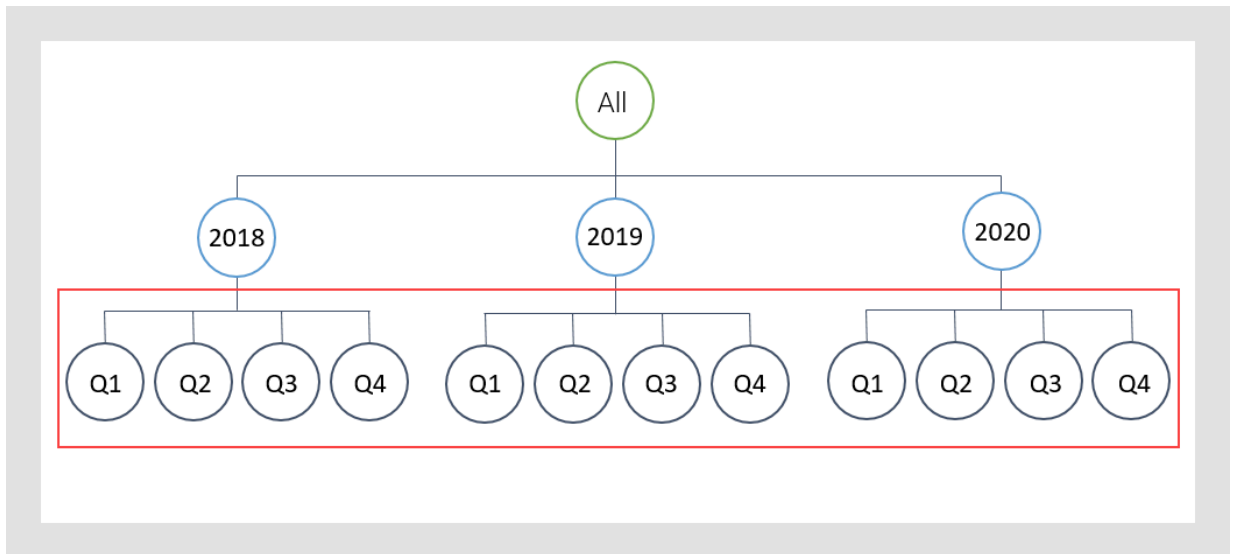


The topics below introduce the different node navigation methods. The expressions used assume that in-memory data is used.

### All()

You use the All() navigation method in an expression, when you want to reference all the nodes on an axis.

This is exemplified by the red rectangle below.



Referencing all the nodes within the rectangle means that the slicing into the 12 nodes in fact will be ignored. Consequently, the All() method is not very useful in itself. However, it is often used in combination with other methods or aggregations, for example, when calculating percentages of a whole. See the examples below.

## Examples



The expressions in the examples assume that in-memory data is used.

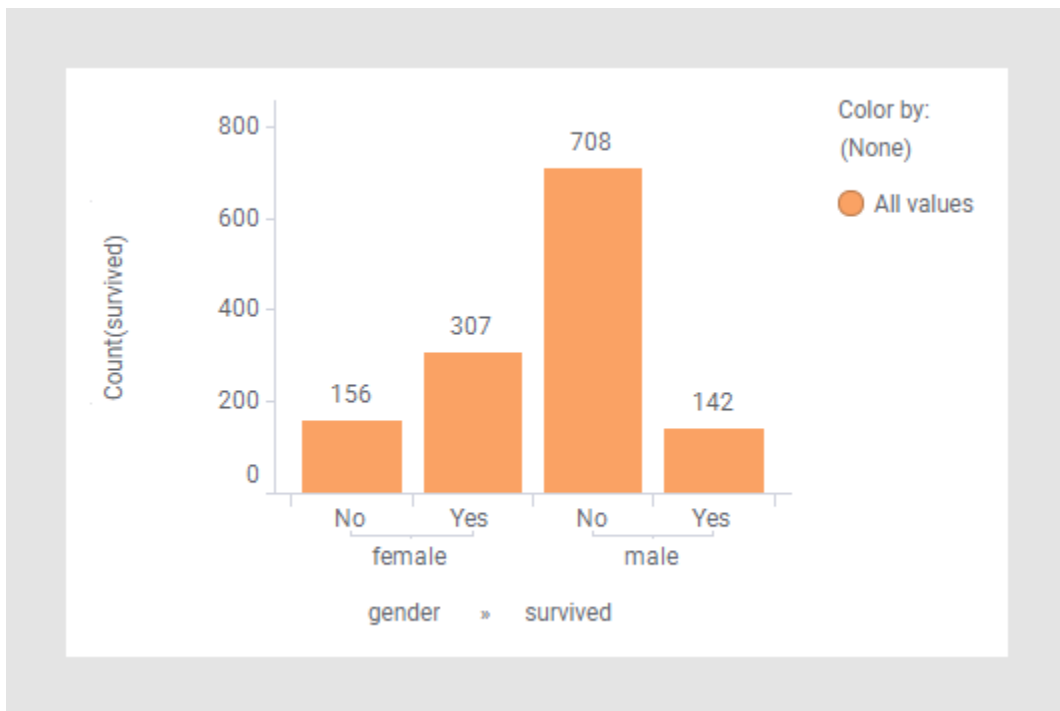
In the following examples, survival statistics from the Titanic catastrophe are used.

name	gender	survived
Allen, Miss Elisabeth Walton	female	Yes
Allison, Miss Helen Loraine	female	No
Allison, Mr Hudson Joshua Creighton	male	No
Allison, Mrs Hudson J.C. (Bessie Waldo Daniels)	female	No
Allison, Master Hudson Trevor	male	Yes
Anderson, Mr Harry	male	Yes

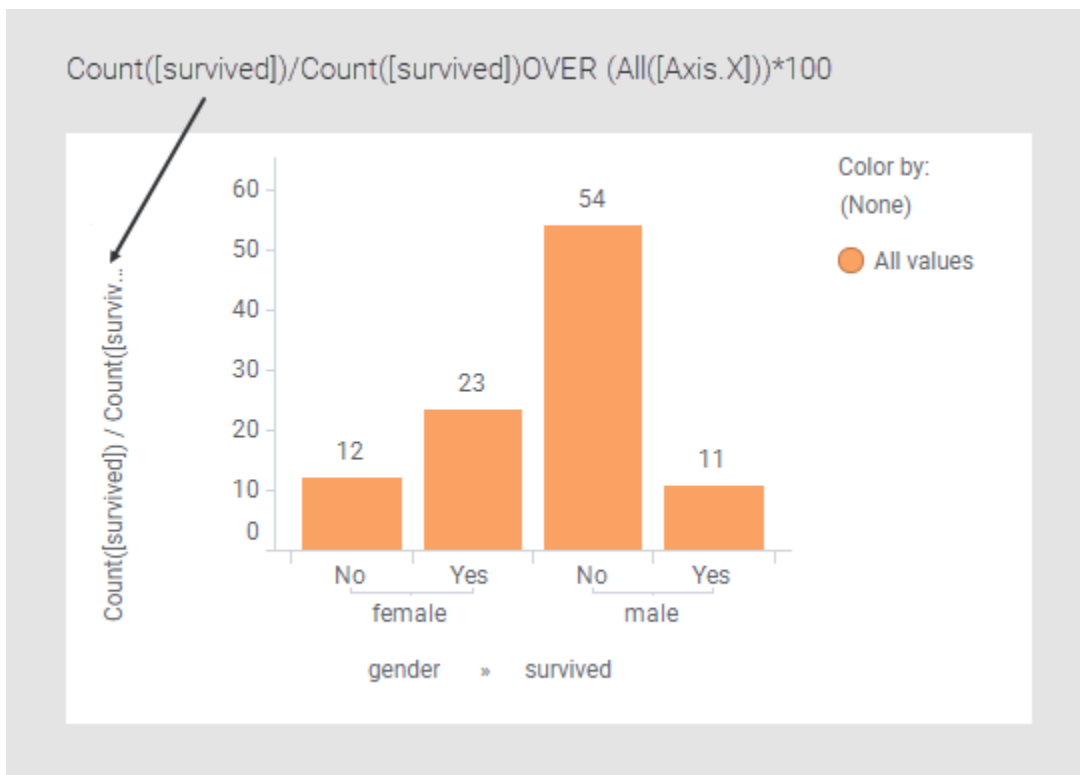
### Example (only the category axis is used for slicing the data)

The bar chart displays the number of women and men, who died and survived in the catastrophe.





Assume you want this information expressed in percent instead of absolute numbers. [Add the following expression](#) on the value axis to calculate the different parts of the whole:

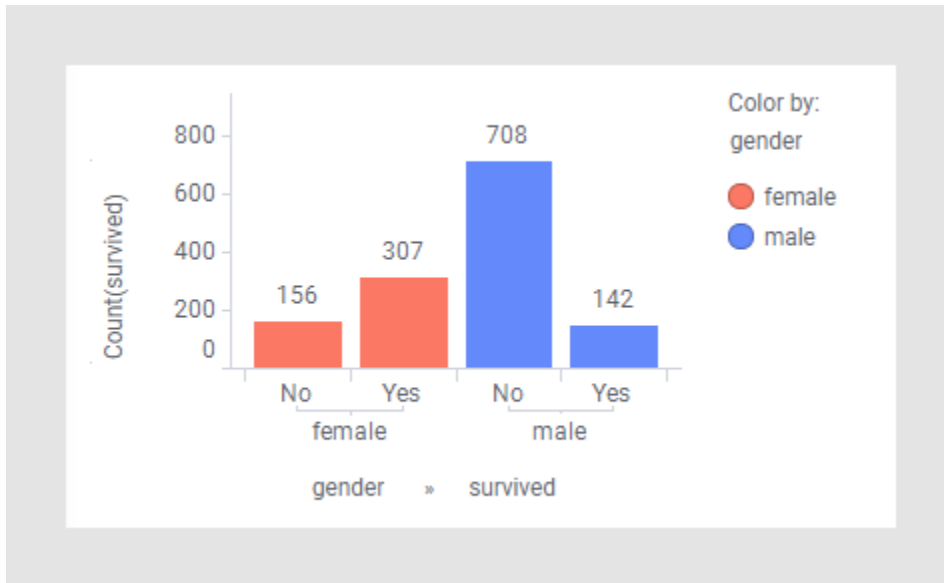


A recalculation of each marker (here each bar is a marker) is performed. For example, the second bar from the left shows that about 23% of the Titanic passengers were women who survived ( $307 / (156 + 307 + 708 + 142)$ ).

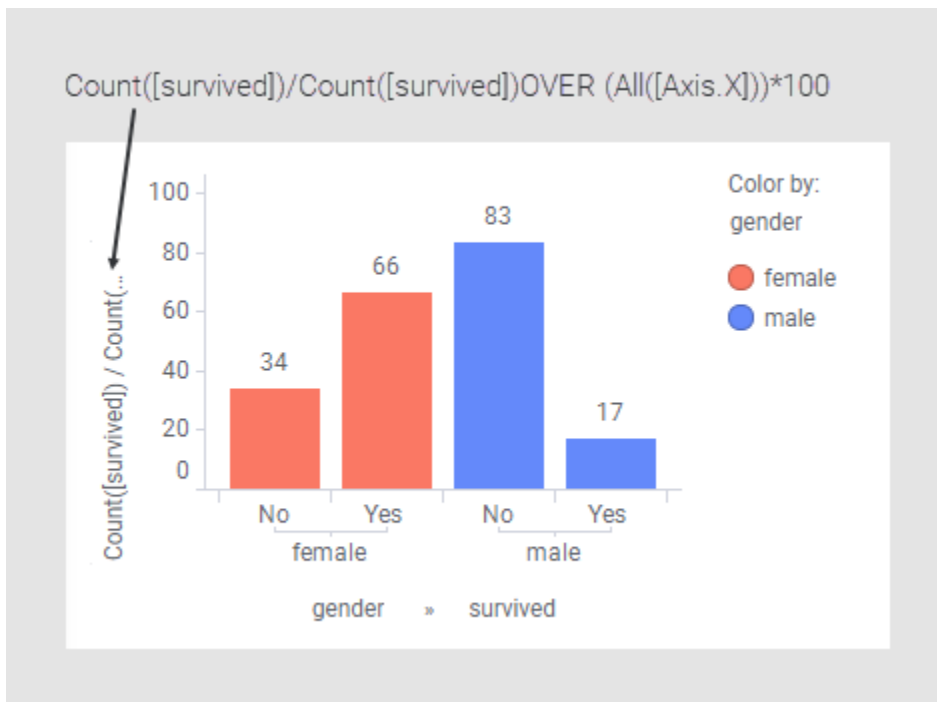
In this example, the category axis is used for slicing. In the next example, the data is sliced using the color axis as well.

### Example (the category axis and the color axis are used for slicing the data)

The bar chart below displays the same data as the bar chart at the top with the addition that the color axis is used to color the bars by gender.



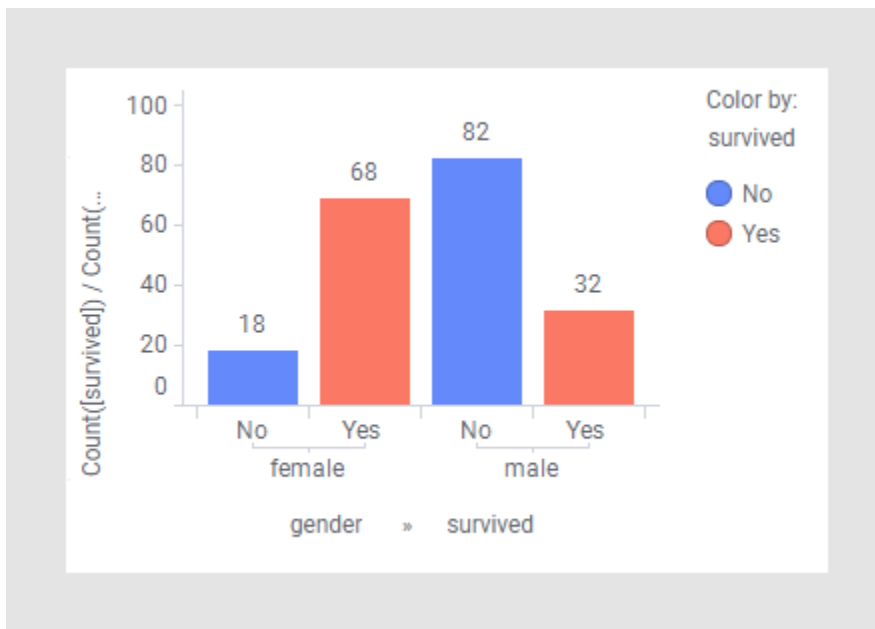
When adding the same expression to the value axis as used above, note the difference:



The percentages add up to 100% within each color instead. This is because the `OVER (All([Axis.X]))` part of the expression declares that the slicing on the category axis should be ignored, but the slicing specified on the color axis is still taken into account. The effect is that the category axis slicing is certainly ignored, but within each of the colors.

For example, 66% of all females survived ( $307 / (307 + 156)$ ).

Also note the difference when specifying the 'survived' column on the color axis:

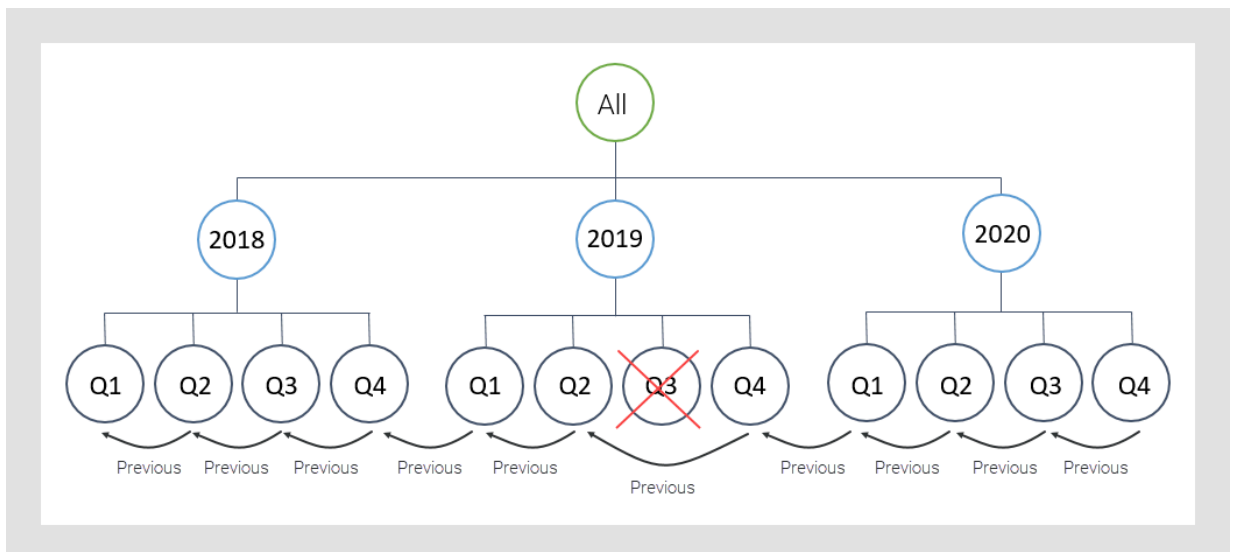


The same expression on the value axis gives another result. Here you can, for example, see that 68% of all survivors were women ( $307/(307+142)$ ).

### Previous()

You use the Previous() navigation method in an expression, when you want to reference the previous node within the same level of the hierarchy.

This is exemplified below.



In case the value preceding the current node is not available (such as the Q3, 2019 value above), the method looks for the very previous, not missing, value (in this case Q2, 2019).

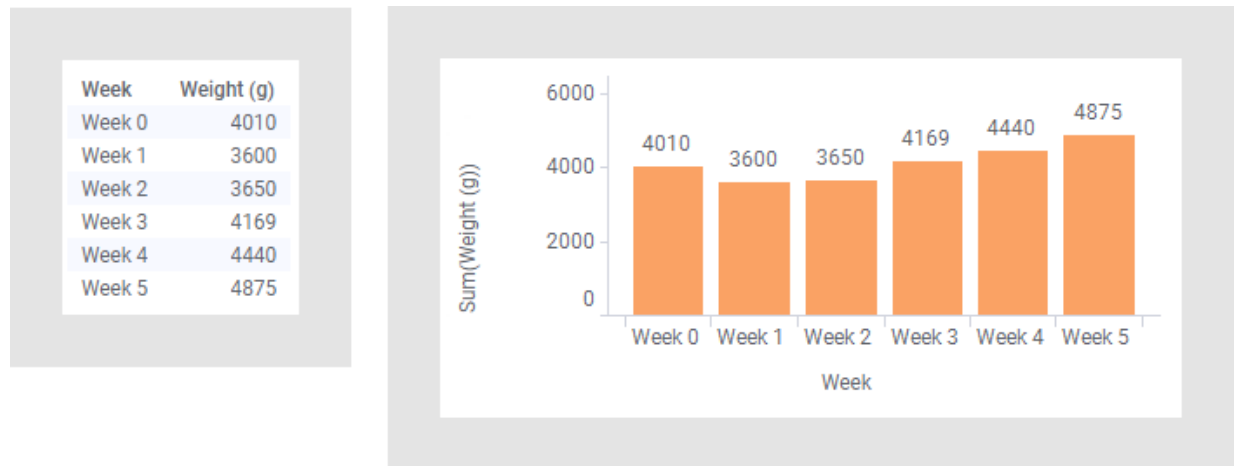
This method is commonly used to calculate differences, for example changes in the current quarter over the previous quarter as illustrated above.

## Example

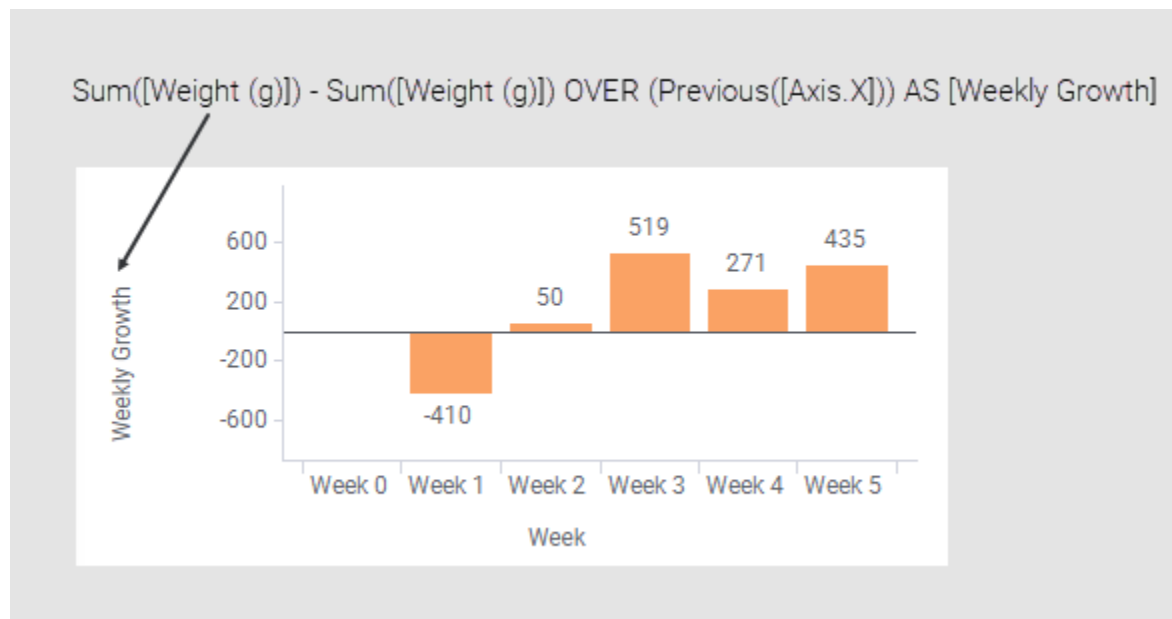


The expression in the example assumes that in-memory data is used.

The data table lists weekly measured weights for a newborn child, and the weights are shown in a bar chart.



Assume you are interested in the child's weekly growth. Then you can apply the following expression on the value axis:



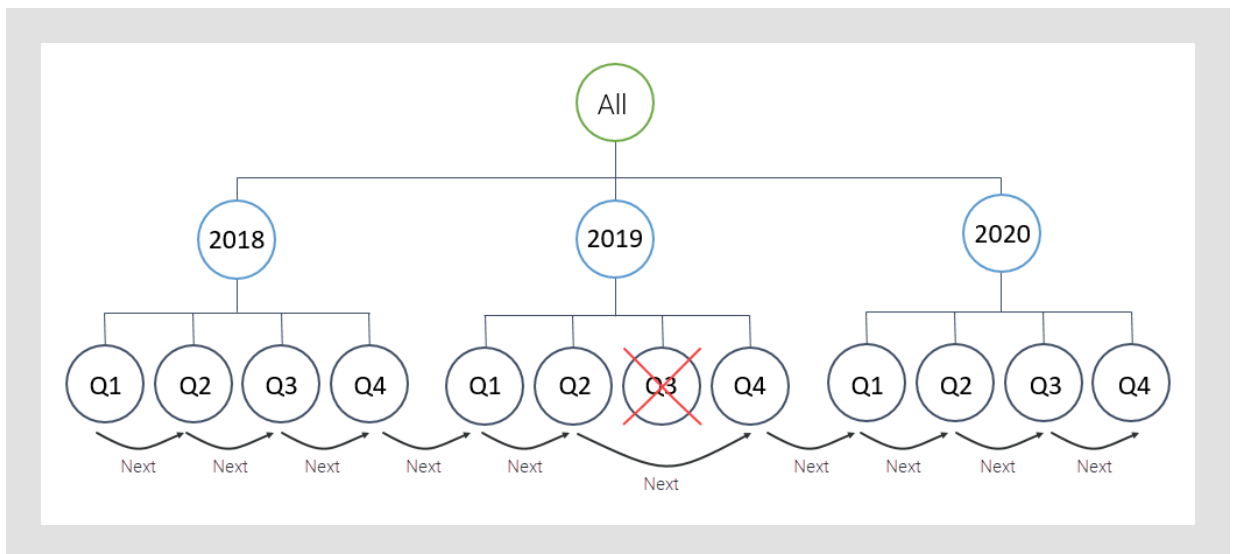
The expression calculates the differences week by week, that is, the value of each bar is calculated by subtracting the value of the previous bar. The bar furthest to the left is dropped, because there is no previous value to reference.

The same example is used in the [Next\(\)](#) topic. Depending on whether Next() or Previous() is used, the resulting bars are presented differently.

## Next()

You use the Next() navigation method in an expression, when you want to reference the next node within the same level of the hierarchy.

This is exemplified below.



If the value following the current node is not available (such as the Q3, 2019 value above), the method looks for the very next, not missing, value (in this case Q4, 2019).

The method is commonly used to calculate differences, for example, changes in the current quarter over the following quarter as illustrated above.

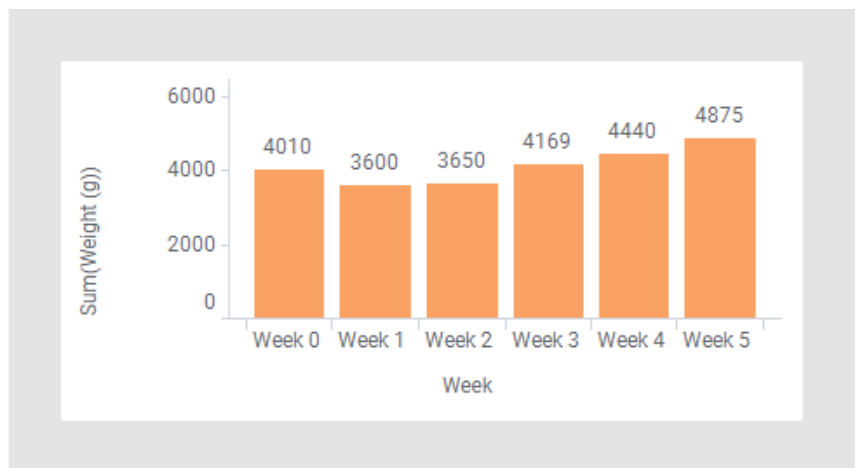
### Example



The expression in the example assumes that in-memory data is used.

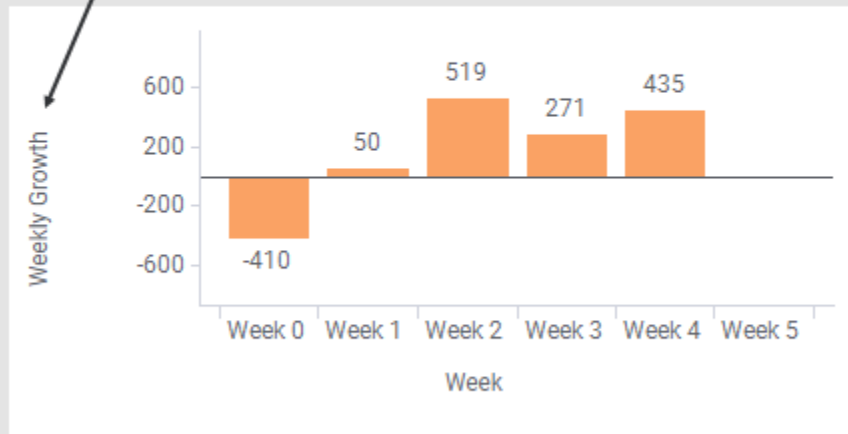
The data table lists weekly measured weights for a newborn child, and the weights are shown in a bar chart.

Week	Weight (g)
Week 0	4010
Week 1	3600
Week 2	3650
Week 3	4169
Week 4	4440
Week 5	4875



Assume you are interested in the child's weekly growth. Then you can apply the following expression on the value axis:

Sum([Weight (g)]) OVER (Next([Axis.X]))- Sum([Weight (g)]) AS [Weekly Growth]



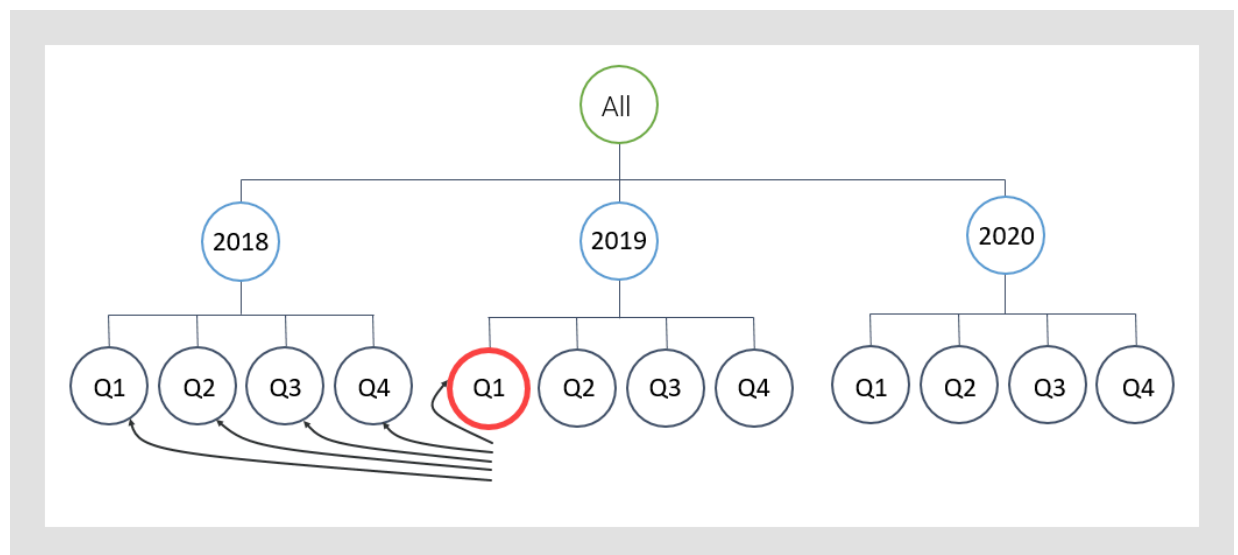
The expression calculates the differences week by week, that is, the value of each bar is calculated by subtracting the value from the value of the following the bar. The bar furthest to the right is dropped, because there is no following value to reference.

The same example is used in the [Previous\(\)](#) topic. Depending on the method used, the resulting bars are presented differently.

### AllPrevious()

You use the AllPrevious() navigation method, when you want to reference the current node along with all previous nodes in the level of the hierarchy.

This is exemplified below.



If, for example Q1 2019 is the current node, the use of the AllPrevious() navigation method will reference five nodes, Q1-Q4 2018, and Q1 2019.

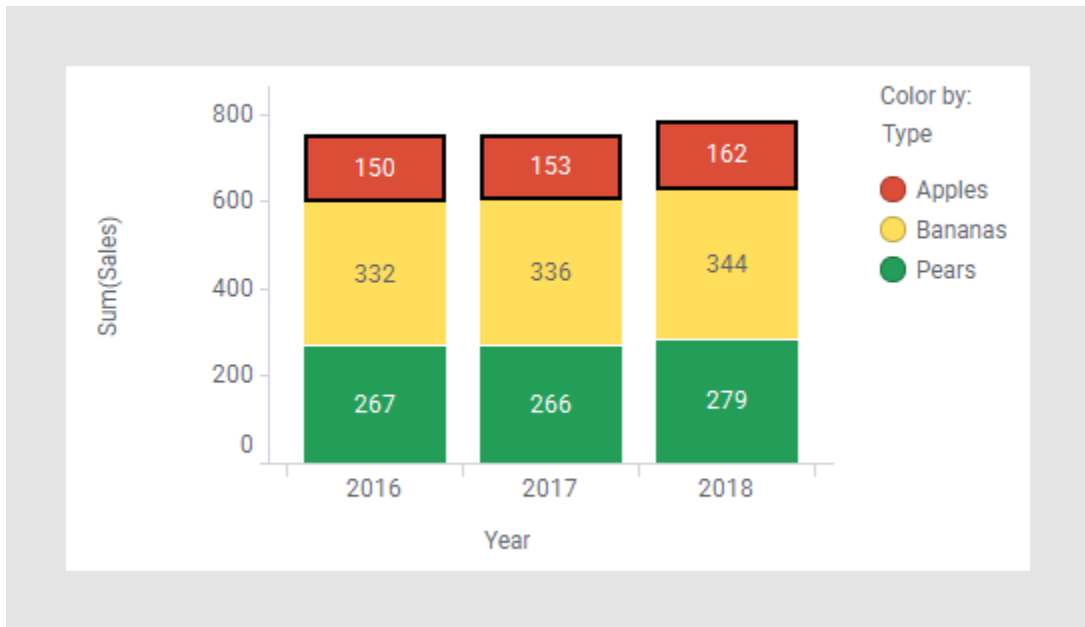
The AllPrevious method is, in particular, used when creating cumulative sums.

## Example

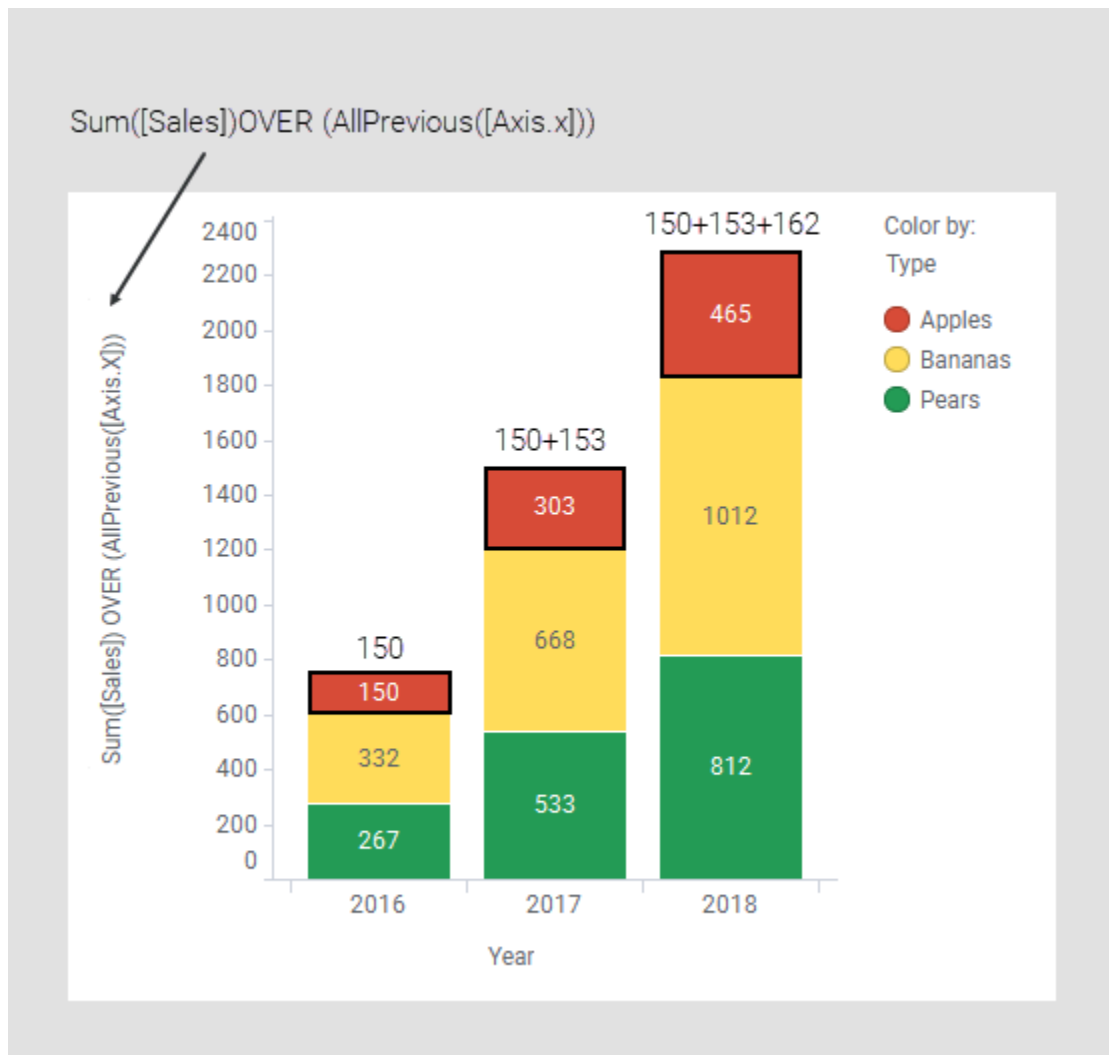


The expression in the example assumes that in-memory data is used.

The yearly sales figures for different fruit types are presented in the bar chart, that is, the data is sliced per Type of fruit and per Year.



Assume you want to examine the cumulative sum of sales as it grows since the start in 2016. Then you can apply the following expression on the value axis:



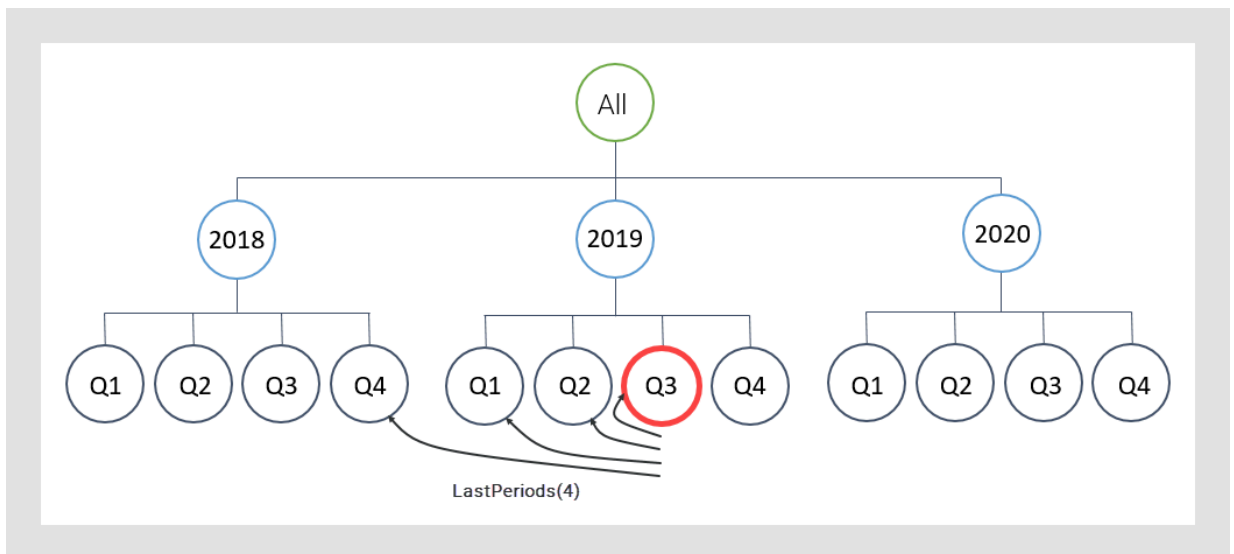
The expression is applied on each slice of the data. For example, 2017, the sales of apples is 303 in total since 2016, and 2018, it is 465.

### LastPeriods()

You use the LastPeriods() navigation option, when you want to reference a set of nodes, prior to and including the current node within the same level of the hierarchy. It is up to you to specify how far to look backwards in time.

For example, if you specify LastPeriods(4), as illustrated below, the method will include the current node and the three prior nodes.





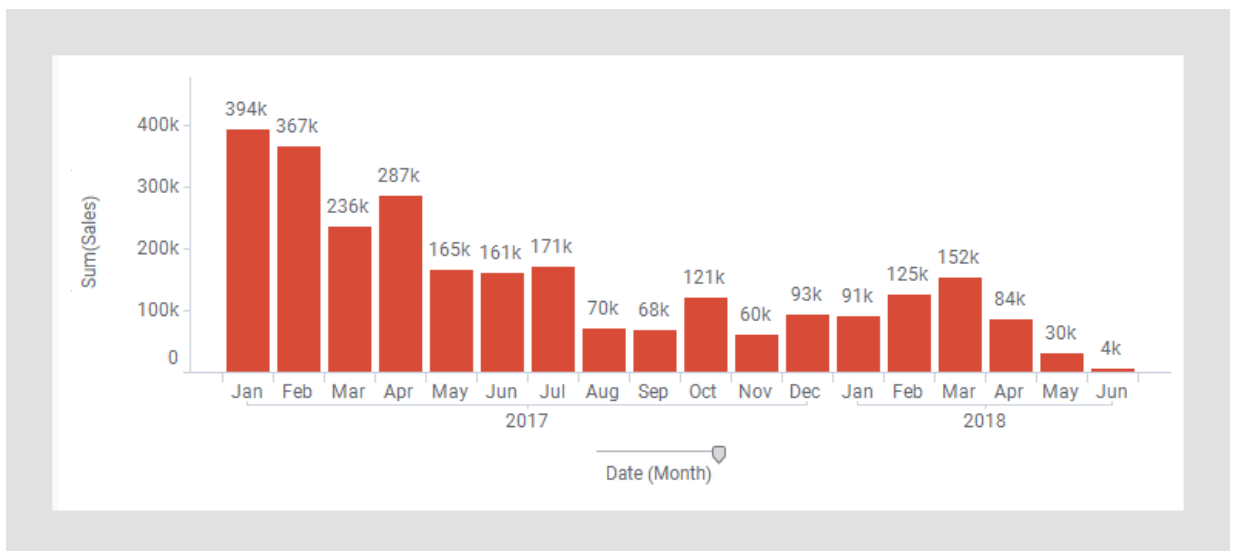
The method is useful in answering common business questions, where information backwards in time is requested, for example, when calculating averages moving over time.

### Example

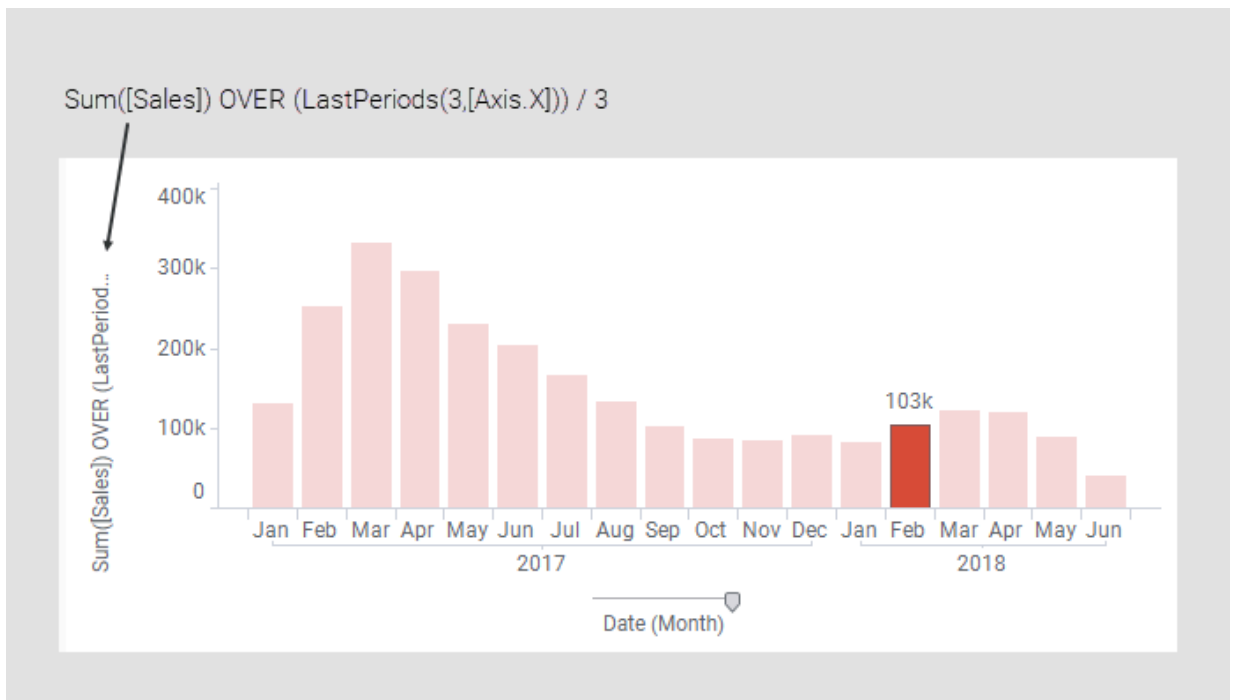


The expression in the example assumes that in-memory data is used.

The bar chart shows monthly sales values for a period of time. Assume you want to view trends in average sales values, looking backwards three months at a time.



Apply the expression below. It sums the sales for three nodes (the current node and its prior two nodes), and divides the sum by 3, that is, calculates the moving average that spans over three months.



For example, the marked February bar (103k) is the average of sales in three months, December 2017, January 2018, and February 2018  $((93k+91k+125k)/3)$ .



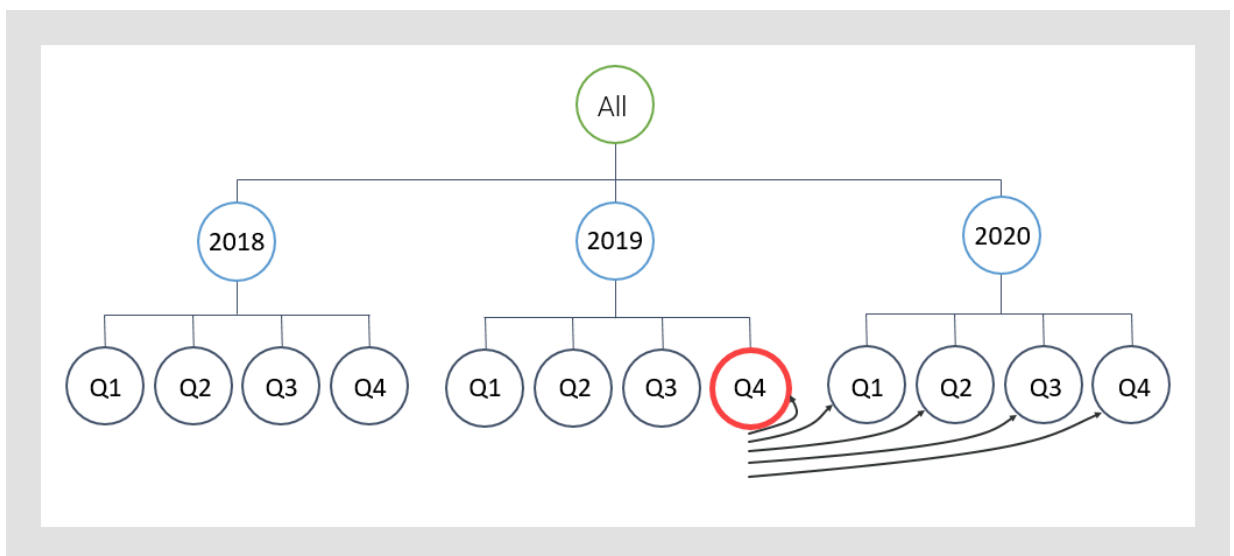
The following expression would have returned the same result:

```
(Sum([Sales]) + Sum([Sales]) OVER (PreviousPeriod([Axis.X])) + Sum([Sales])
OVER (NextPeriod([Axis.X]))) / 3
```

### AllNext()

You use the AllNext() navigation method, when you want to reference the current node along with the nodes that follow in the level of the hierarchy.

This is exemplified below.



If, for example Q4 2019 is the current node, the use of the AllNext() navigation method will reference five nodes, Q4 2019, and Q1-Q4 2020.

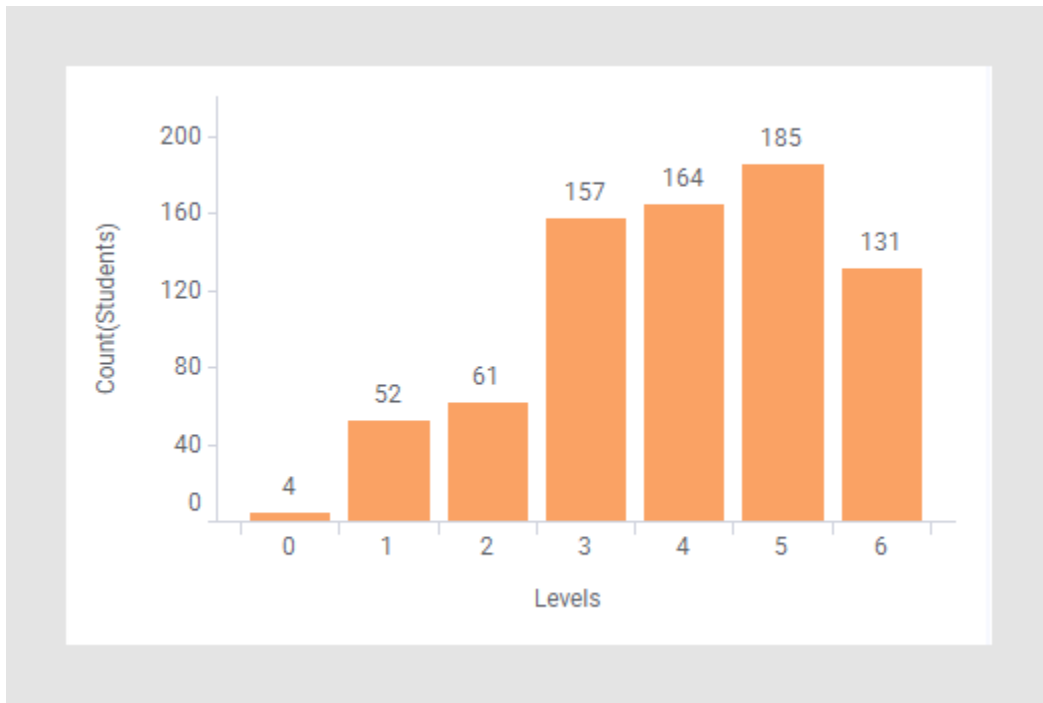
The AllNext() and AllPrevious() methods are similar, and they are useful when accumulating values in the nodes.

### Example



The expression in the example assumes that in-memory data is used.

The bar chart displays how many students that reached various levels in a test. For example, 52 students reached level 1, and 131 reached level 6.



Assume you would like to know the percentage share of students that managed to reach, say, level 5 or higher. Apply the expression below, which uses the AllNext() as well as the All() method, on the value axis.

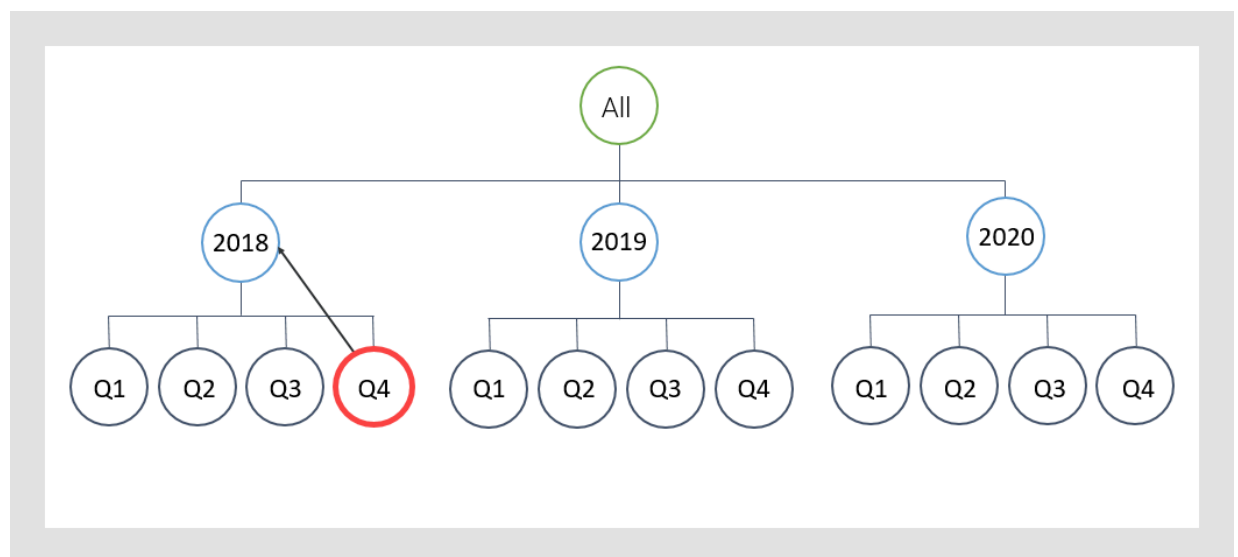
$\text{Count}([\text{Students}]) \text{ OVER } (\text{AllNext}([\text{Axis.X}])) / \text{Count}([\text{Students}]) \text{ OVER } (\text{All}([\text{Axis.X}])) * 100$



To explain what result the expression returns for each node, let us use level 5 as the current node to be re-evaluated. The first OVER expression,  $\text{Count}([\text{Students}]) \text{ OVER } (\text{AllNext}([\text{Axis.X}]))$ , returns the number of students who reached level 5 or higher, that is, 316 (185 + 131), and the other part,  $\text{Count}([\text{Students}]) \text{ OVER } (\text{All}([\text{Axis.X}]))$ , returns the total number of students, 754. So, 316 of the 754 passed at least level 5, that is, approximately 41,91% (316/754).

### Parent()

You use the `Parent()` navigation method, when you want to reference a parent node in the hierarchy. This is exemplified below.



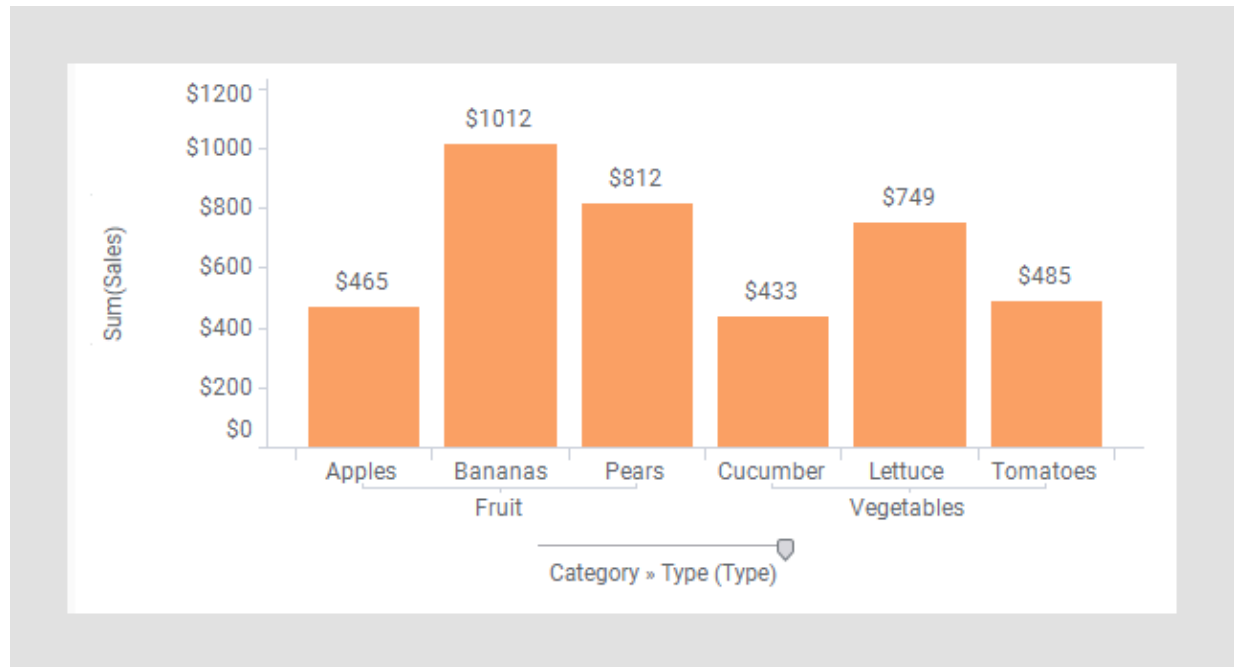
From the current node, you navigate up one level of the hierarchy. If the data is not hierarchical, or the current node is at the top level, the `Parent()` method works in the same way as the `All()` method.

## Example

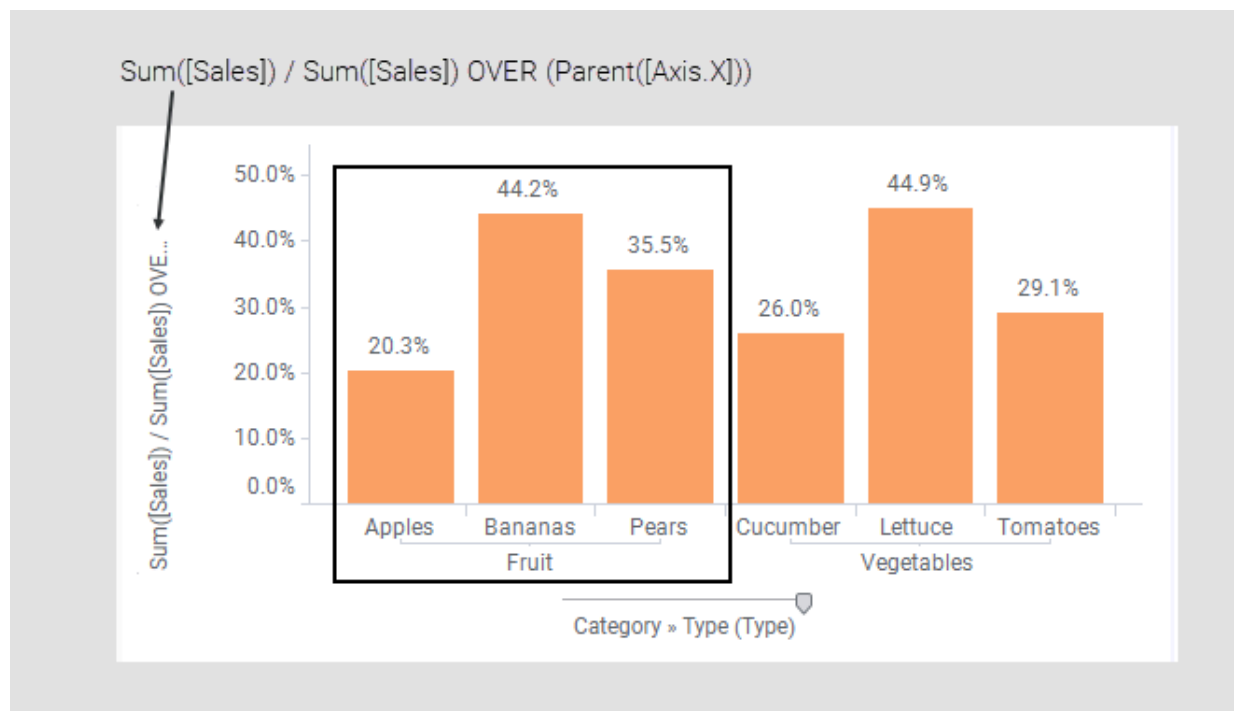


The expression in the example assumes that in-memory data is used.

The bar chart below presents absolute sales figures for a number of products. The products have been structured hierarchically on the category axis.



Assume you want to display each fruit type's share of the total fruit sales, and each vegetable's share of the total vegetables sales. Then you need to reference the parent nodes. Apply the expression below (and format the values as percentage on the Value axis):

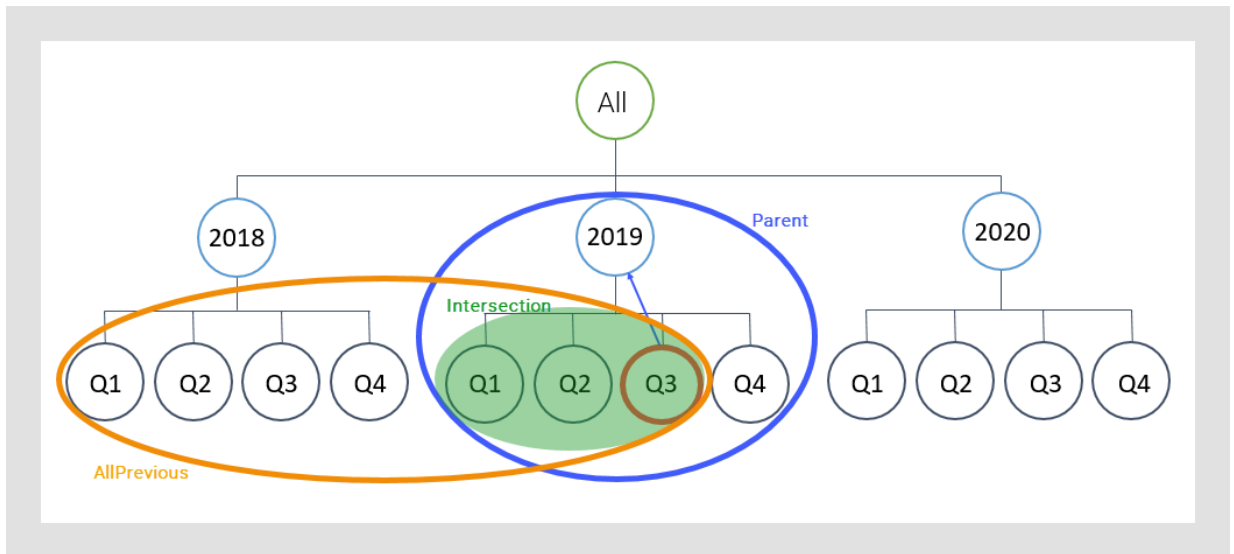


The expression divides each fruit's sales amount by the total fruit sales, and each vegetable's sales amount by the total vegetables sales. For example, from the Bananas sales, the Parent() method navigates up to the Fruit level to retrieve the total sum of fruit. The ratio is  $1012/(465+1012+812) \approx 44.2\%$ .

## Intersect()

You use the Intersect() navigation method, when you want to reference the nodes that are the result of the intersection of two or more node navigation methods.

This is exemplified below.



The image illustrates, with Q3 as the current node, the intersection of the AllPrevious() and the Parent() navigation methods. The nodes that are found by both the methods are Q1-Q3, 2019.

No matter which quarter is the current node, the intersection will result in a period that begins with Q1 and ends with the quarter that is the current node. This means that a typical use case for the Intersect() method is a Year to Date calculation (see the example below).

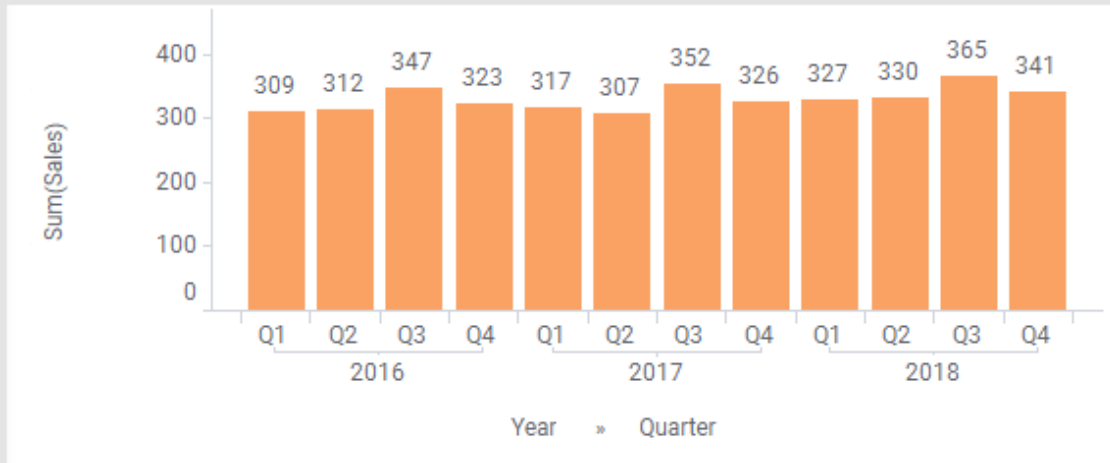
## Examples



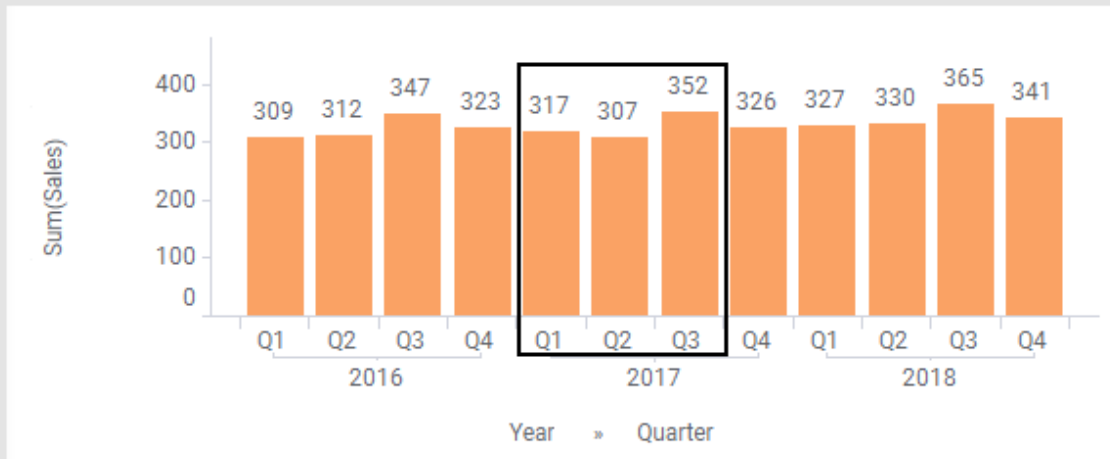
The expressions in the examples assume that in-memory data is used.

### Example of Year to Date calculation (YTD)

The bar chart shows quarterly sum of sales during 2016-2018.



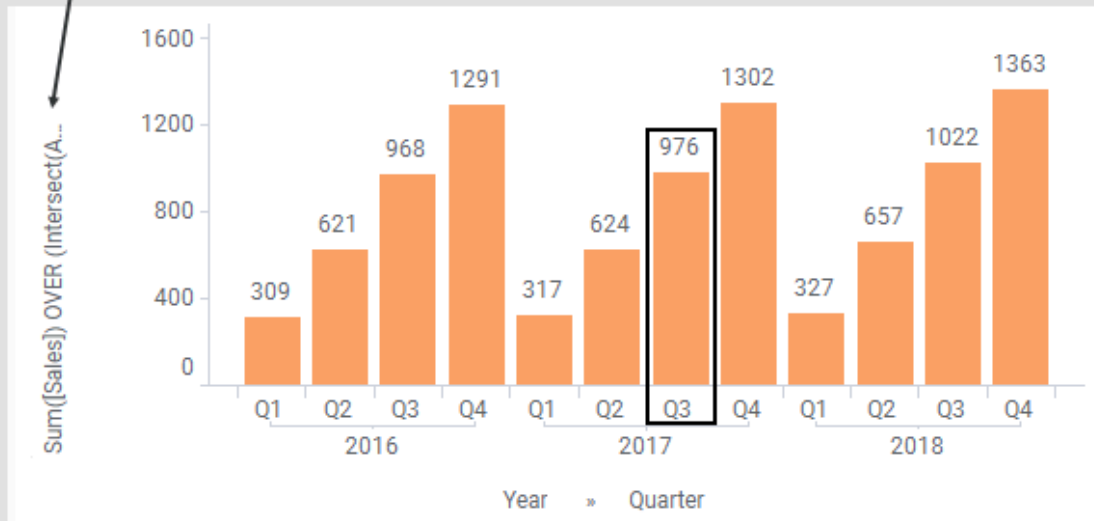
Assume you are interested in the cumulative sums for each of the years. If you use the AllPrevious() navigation method, you will sum all the previous months. But in Q1 2017, you want to start the summing all over again, and disregard the summing of the quarters in the previous year, 2016.



For example, the cumulative sum Q3 2017 should be 976 (317+307+352). To reference only these three bars, the Intersect() method is handy. The AllPrevious() method would include Q1-Q4 2016 and Q1-Q3 2017 in the summing, and the Parent() method would include all four quarters of 2017. The intersection of these two navigation modes, would return exactly the bars in question.

Apply the expression below where Intersect() combines the two methods, and view the result for, for example, Q3 2017:

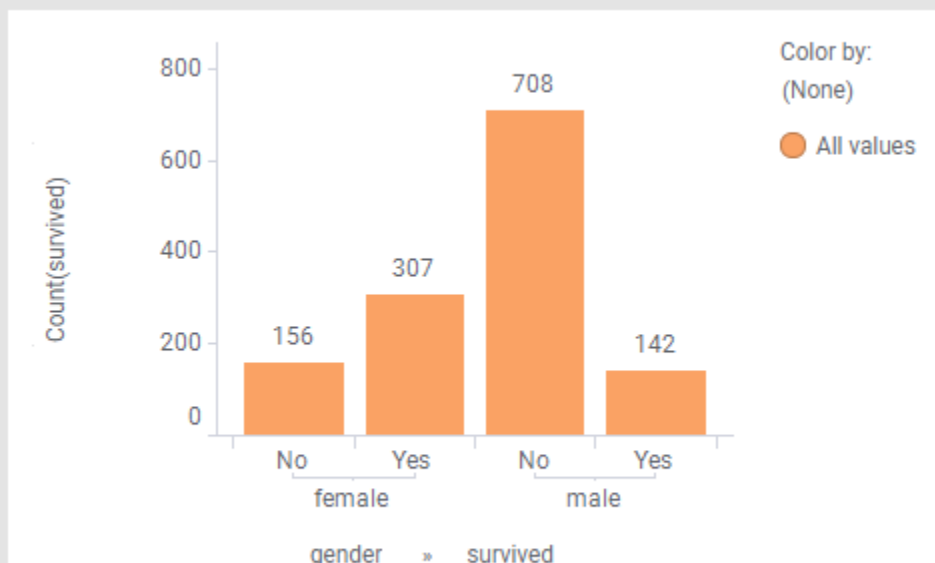
Sum([Sales]) OVER (Intersect(AllPrevious([Axis.X]),Parent([Axis.X])))



### Example of using Intersect() when data is split by more than one axis

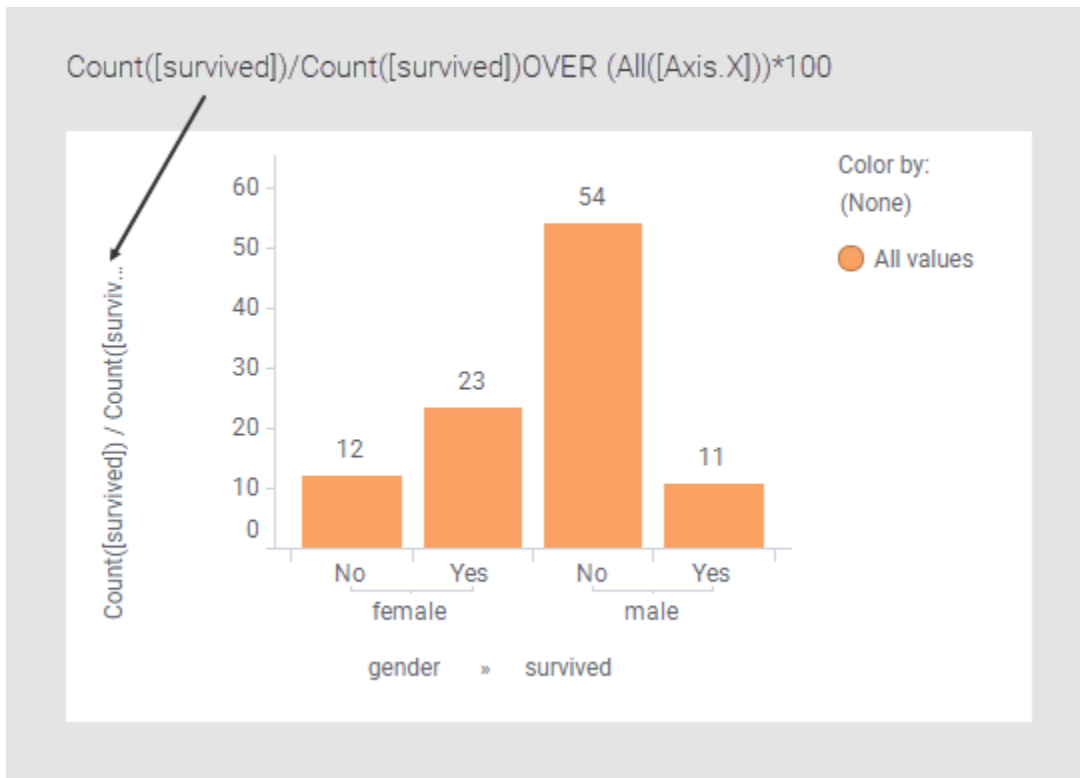
Each time you slice the data using various axes, the data is split into smaller and smaller slices, represented by more and more markers. When you apply an expression, it will work on the smallest slice of data. In the example above, the data is split solely by the category axis. But what happens if you split, for example, also by the color axis? An example will show you.

The bar chart below displays survival statistics from the Titanic catastrophe. The data is split solely by the category axis.

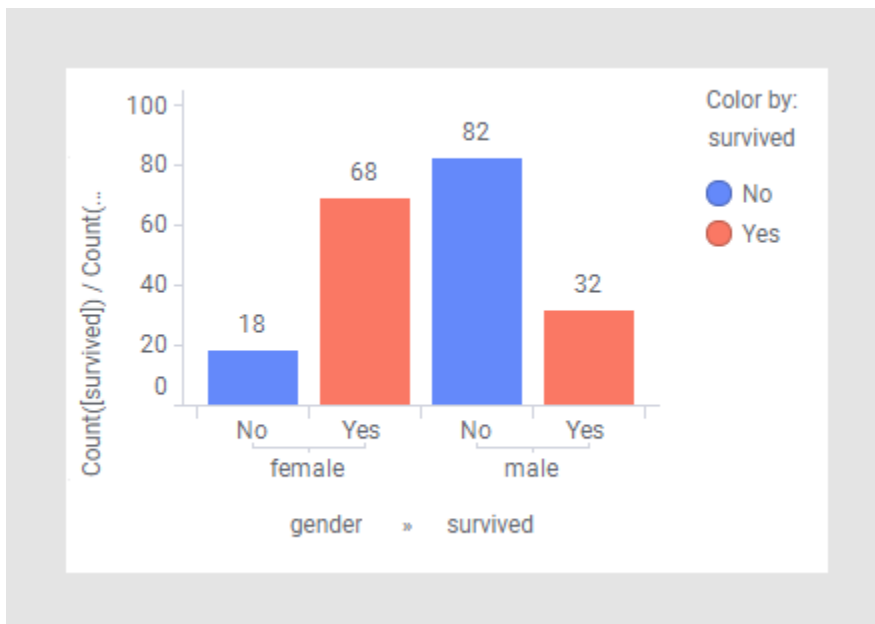




The expression in the next image lets you display this information in percent instead of absolute numbers. See also the [All\(\)](#) topic.

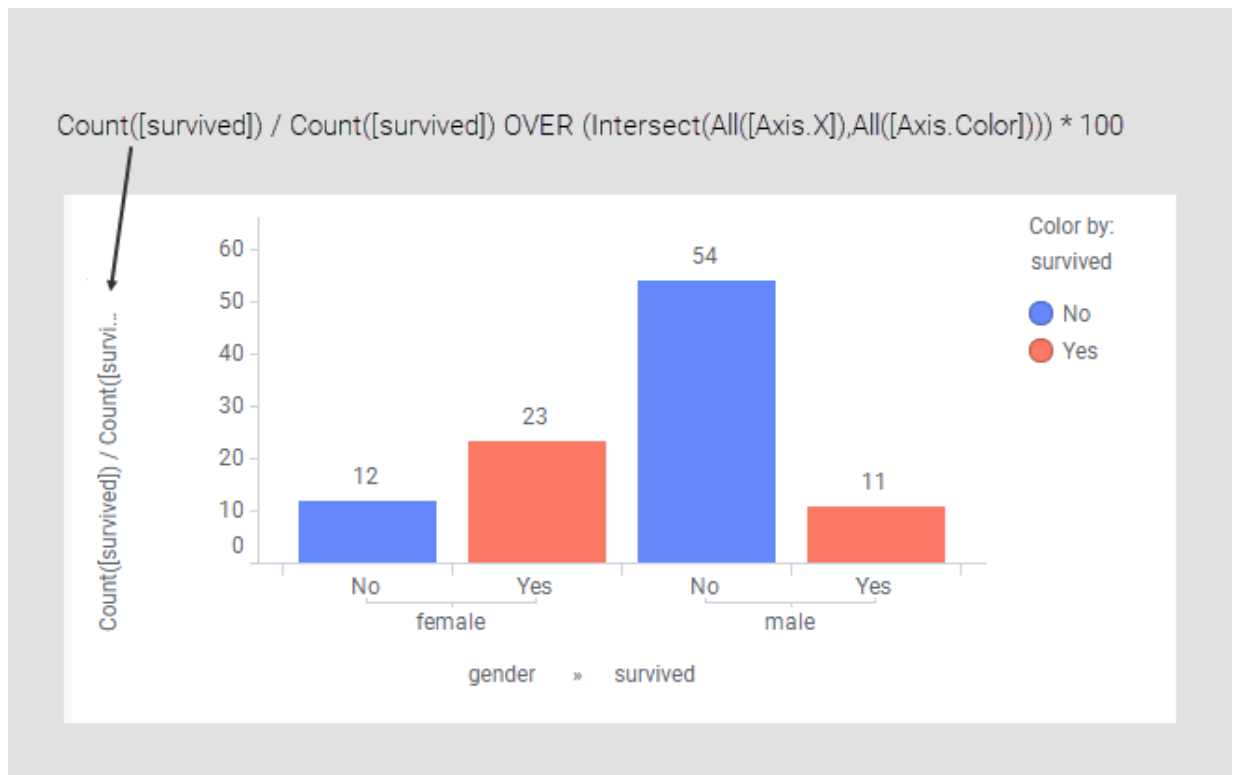


Assume you want to distinguish the bars representing those who survived by setting the Color axis to 'survived' as shown below.



However, note that when doing so, the percentages change! The red and blue bars, respectively, add up to 100% instead of all the four bars. This happens because the expression does not take into account that the data is split into even smaller slices by the color axis. The `...OVER (All([Axis.X]))...` part ignores the slicing on the category axis, but not the slicing on the color by axis.

You can, though, keep the colors and add up all the bars to 100% by using the `Intersect()` method as shown below.

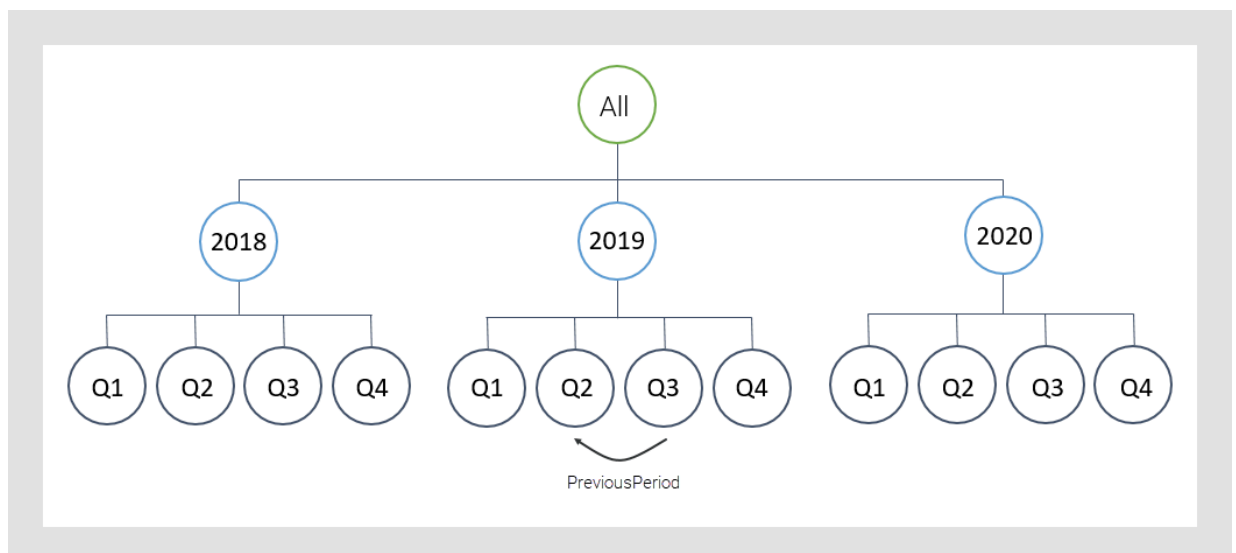


The All([Axis.X]) and the All([Axis.Color]) parts of the expression imply that the slicing on each of these axes should be ignored. The Intersect() method then combines these two parts. Consequently, the value of each bar will be divided by the sum of all the bars.

### PreviousPeriod()

Some node navigation methods are especially suited for hierarchies containing periodic data, first and foremost time hierarchies. Time is structured hierarchically by nature (like year, quarter, month) and the time periods are recurrent. Because the nodes in a time hierarchy are obvious, any nodes that miss data can be detected automatically. One of these navigation methods is PreviousPeriod(), which references the previous node within the same level of the hierarchy. When you use this method in an expression, and any nodes are missing data, the expression will return empty values for such nodes.

The PreviousPeriod() method is exemplified below.

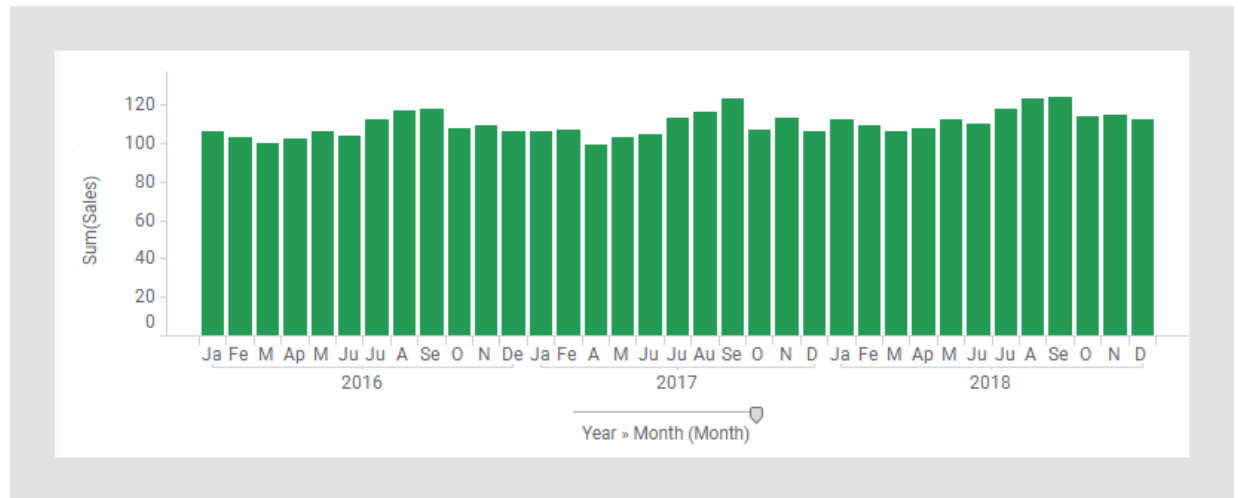


## Example

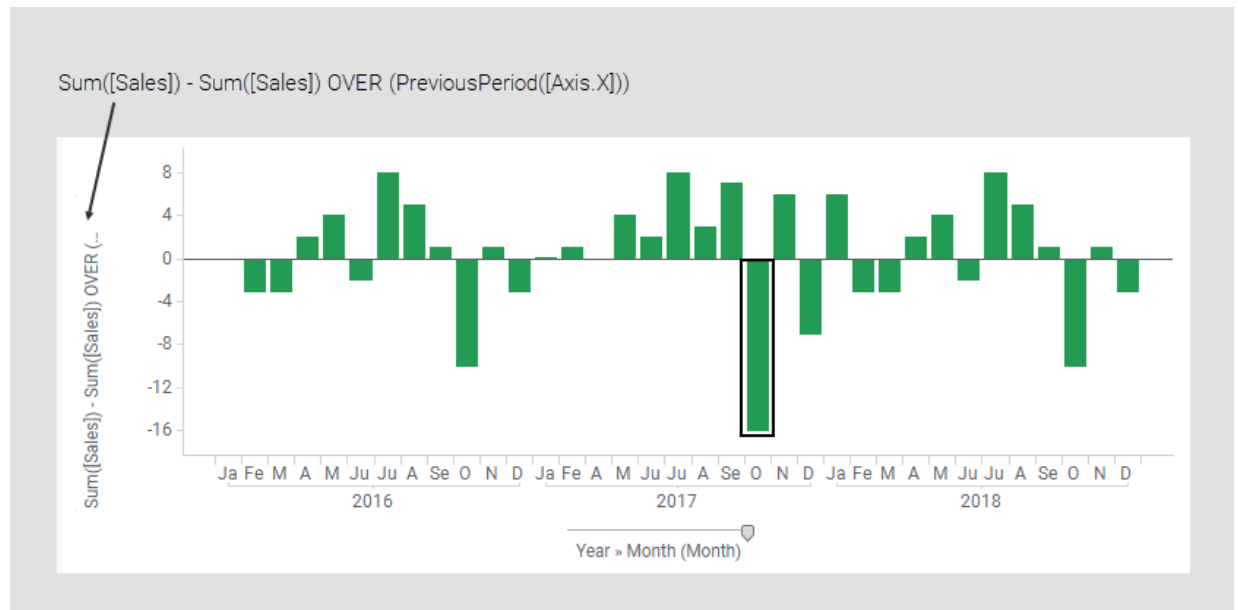


The expression in the example assumes that in-memory data is used.

The bar chart shows sales for a three-year period. The sales figures fluctuate, and it is hard to distinguish where dips occur.



To highlight months with dips compared to the previous month, you can apply the following expression on the value axis:



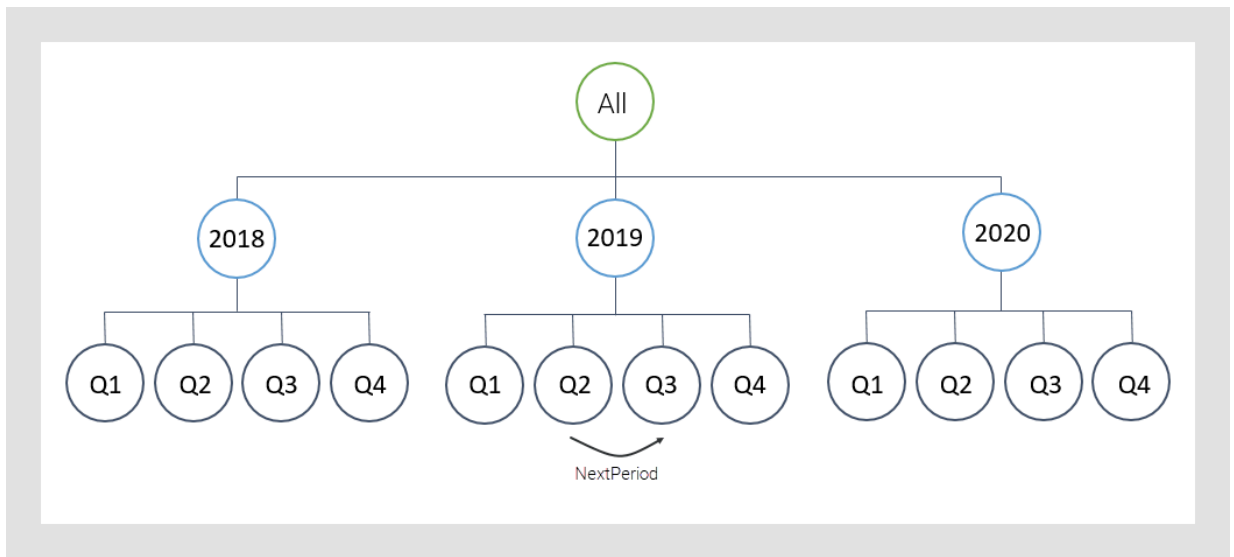
The expression calculates the differences month by month. If the sales figures have decreased compared to the month before, the bars get negative values. Now it is easy to identify, for example, that the deepest dip occurred from September to October, 2017.

## NextPeriod()

Some node navigation methods are especially suited for hierarchies containing periodic data, first and foremost time hierarchies. Time is structured hierarchically by nature (like year, quarter, month) and the time periods are recurrent. Because the nodes in a time hierarchy are obvious, any nodes that miss data can be detected automatically. One of these navigation methods is `NextPeriod()`, which references

the next node within the same level of the hierarchy. When you use this method in an expression, and any nodes are missing data, the expression will return empty values for such nodes.

The NextPeriod() method is exemplified below.

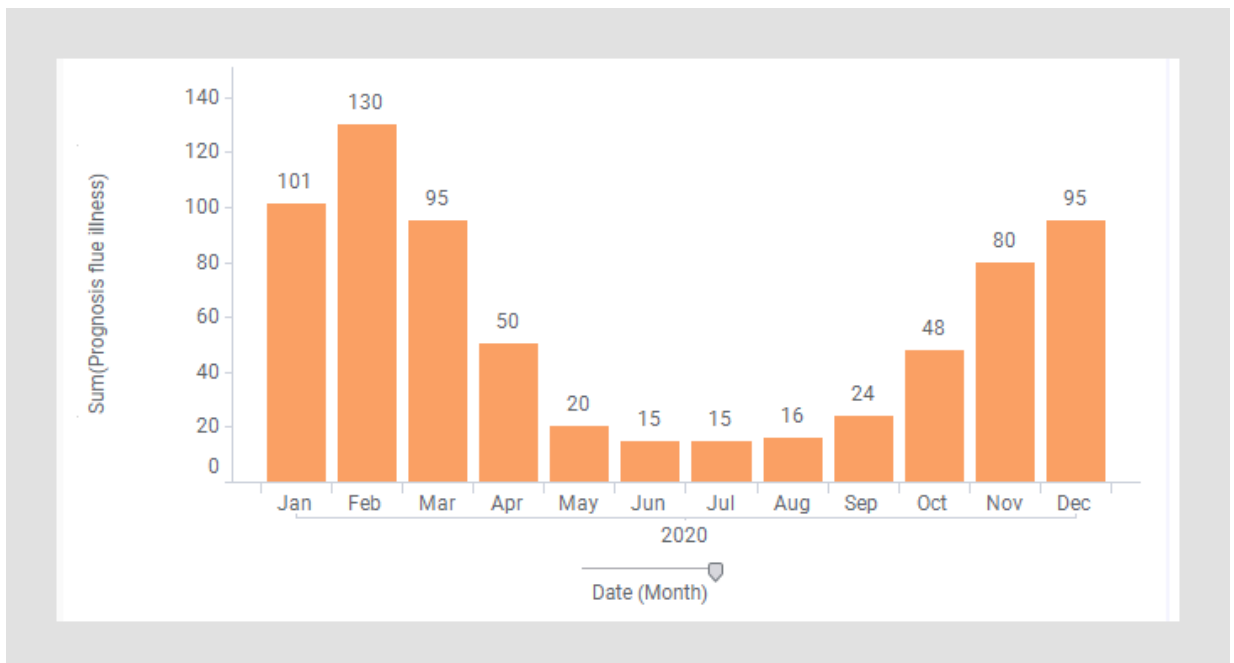


### Example

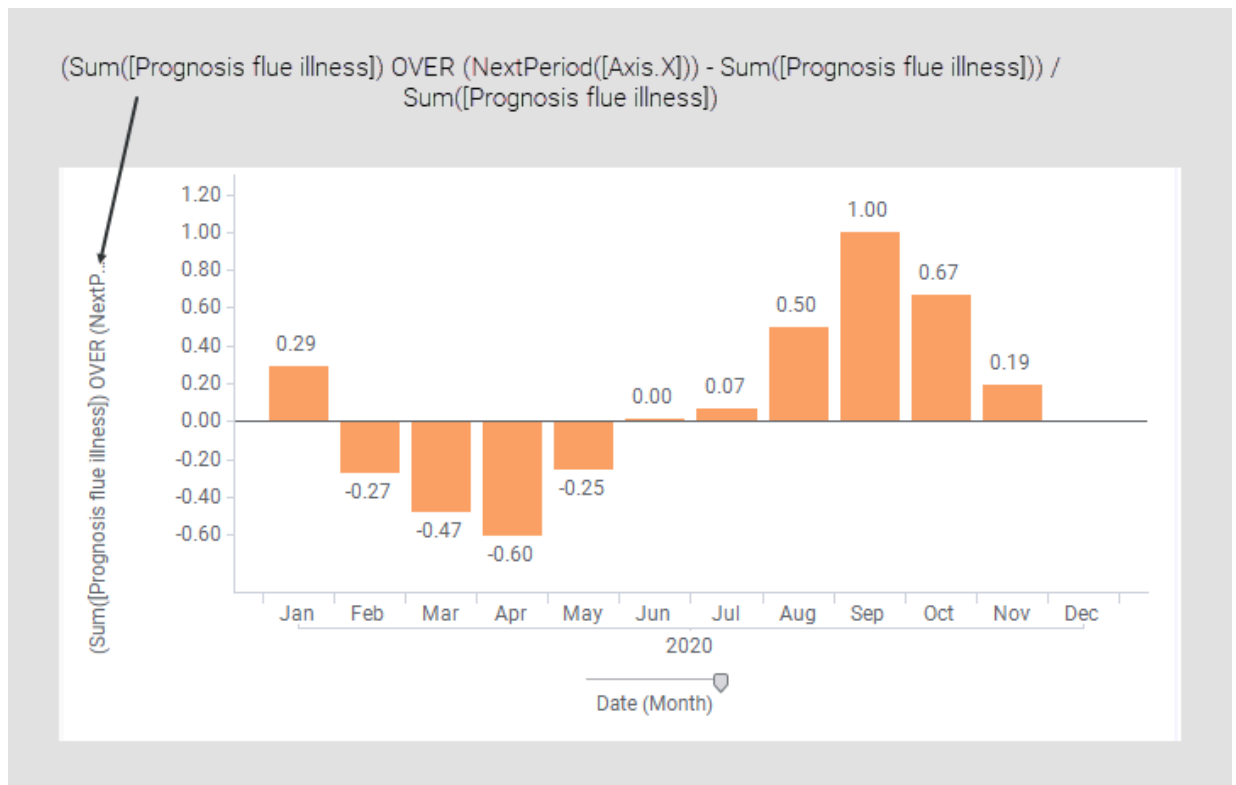


The expression in the example assumes that in-memory data is used.

In January 2020, the falling ill with the flu is forecasted month by month for the entire year, and the prognosis is presented in the bar chart below.



Assume you want the prognosis to show how the relative falling ill develops from one month to the next. Then you can apply the following expression on the value axis.

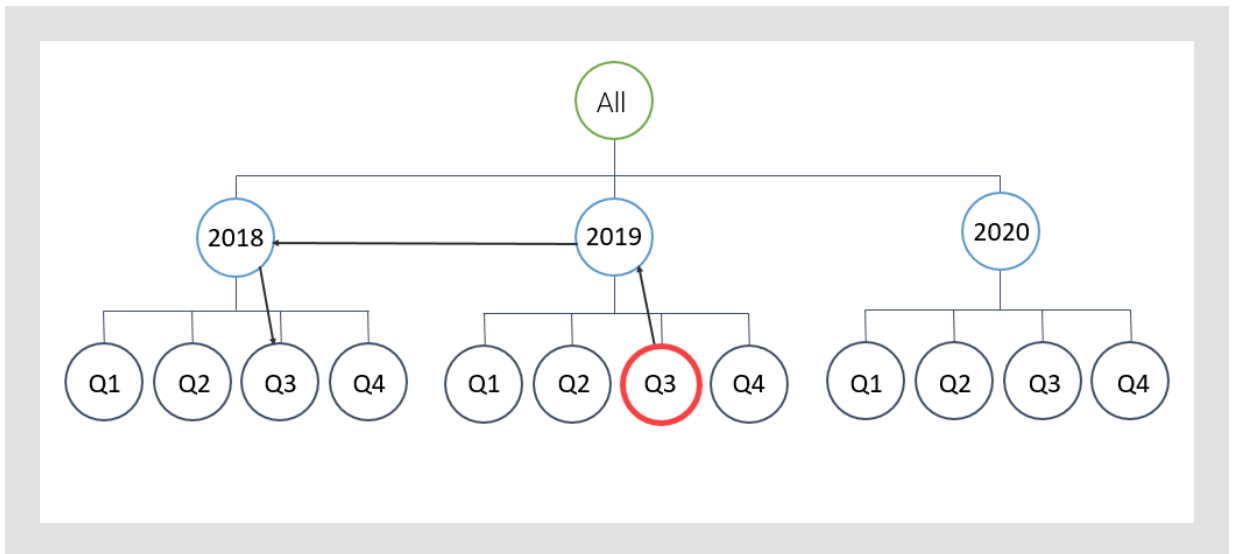


For example, looking at the September bar, the prognosis forecasts that the number of people falling sick will double from September to October (that is, the bar shows an increase of 100%). In October and November, the absolute number of people forecasted to fall sick will still increase, but at a lower growth rate.

### ParallelPeriod()

Some node navigation methods are especially suited for hierarchies containing periodic data, first and foremost time hierarchies. Time is structured hierarchically by nature (like year, quarter, month), and the time periods are recurrent. Because the nodes in a time hierarchy are obvious, any nodes that miss data can be detected automatically. One of these navigation methods is `ParallelPeriod()`, which references the corresponding node in the previous branch of the hierarchy within the same level of the hierarchy. When you use this method in an expression, and any nodes are missing data, the expression will return empty values for such nodes.

The `ParallelPeriod()` method is exemplified below.



This method combines a set of node navigation methods into one method. For example, using Q3, 2019 as the current node, the method navigates up one level in the hierarchy, then navigates to this level's previous node, and then down one level to the node that matches the starting point.



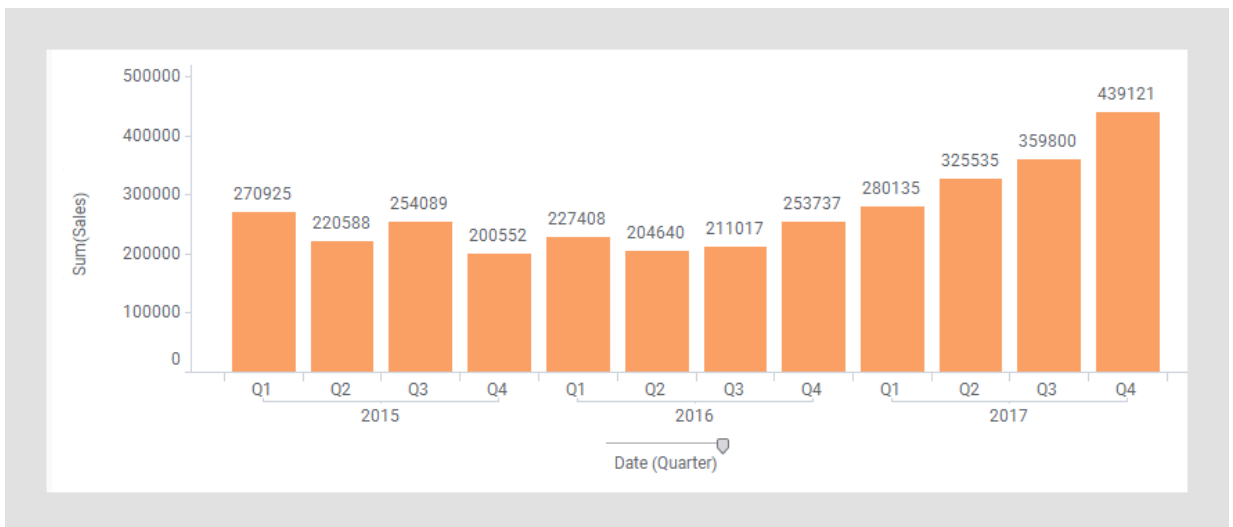
You cannot use the method in a time hierarchy with three levels, years, quarters, and months. This is because the method would not find a matching node. For example, if the starting point is April in Q2 2019, the method navigates from April up to the parent Q2, then back to Q1, but will not find any April there when navigating down.

### Example

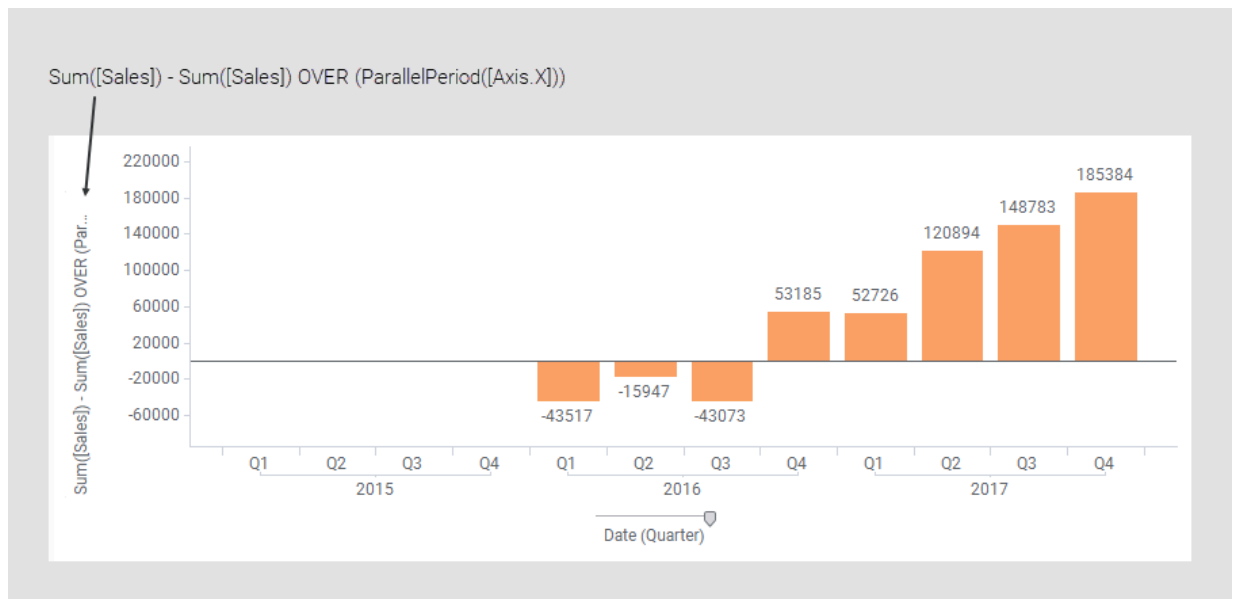


The expression in the example assumes that in-memory data is used.

The bar chart shows quarterly sales during 2015-2017.

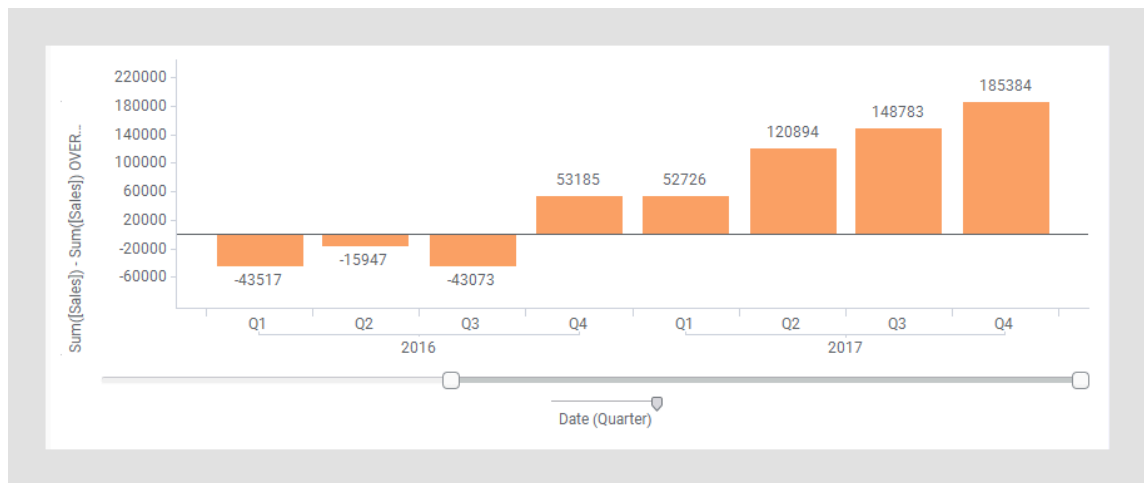


Assume you want to compare the sales in a particular quarter with the sales in the same quarter the previous year, for example, compare sales in Q3 2017 with sales in Q3 2016. Apply the following expression (which calculates the difference in sales in absolute figures):



We can now see that, for example, the sales amount in Q3 2017 exceeded the sales in Q3 2016 by 148783 (359800-211017). In addition, now it becomes easier to identify quarters with sales going down compared to the sales one year ago, as these bars get negative values.

The part furthest to the left is empty, because there are no previous nodes to make a comparison with. If you want to hide this part, [add a zoom slider](#) and drag the slider to a suitable position as shown below.



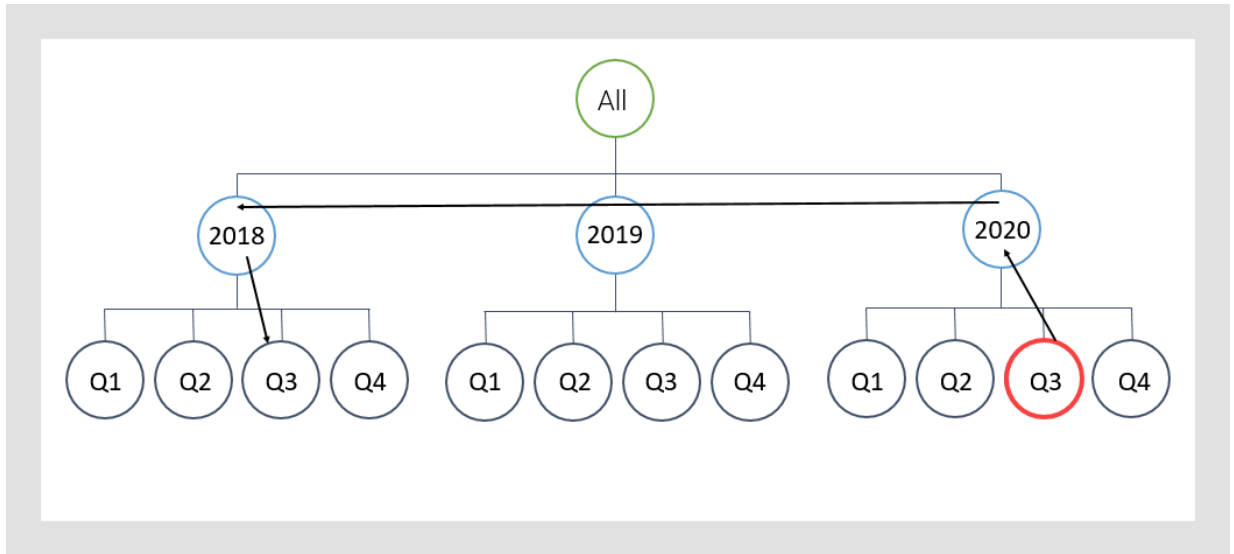
## NavigatePeriod()

Some node navigation methods are especially suited for hierarchies containing periodic data, first and foremost time hierarchies. Time is structured hierarchically by nature (like year, quarter, month), and the time periods are recurrent. One of these navigation methods is `NavigatePeriod()`. This method lets you define your own navigation in a hierarchy, because you can specify four arguments, that is, input values that control which axis to reference, how many levels to move up in the hierarchy, how many steps to move sideways, and how many levels to move down.

As a matter of fact, the `NavigatePeriod()` method can replace many of the other navigation methods. See [Comparing the `NavigatePeriod\(\)` and `Parent\(\)` methods](#).

Moreover, because the nodes in a time hierarchy are obvious, any nodes that miss data can be detected automatically.

The `NavigatePeriod()` method is exemplified below.



This example uses Q3, 2020 as the starting node. Assume you want to reference Q3, 2018. Then you can define a `NavigatePeriod()` expression with three arguments:

```
NavigatePeriod([Axis.X], "Year", -2)
```

The first argument is the axis in question, the second argument navigates up to the Year level, and the third argument navigates sideways at this Year level two steps backwards in time because the value is negative. When a fourth argument is omitted, you automatically return to the same level as the starting point, in this case the corresponding quarter in 2018. You can, if you wish, specify another level than the leaf level by adding a fourth argument.



When specifying a hierarchy level of interest, you can just as well replace the name of the level by the number of steps to move up or down. Using 0 as argument means that you stay on the level. In the image above, `NavigatePeriod([Axis.X], 1, -2)` would have given the same result as `NavigatePeriod([Axis.X], "Year", -2)`.

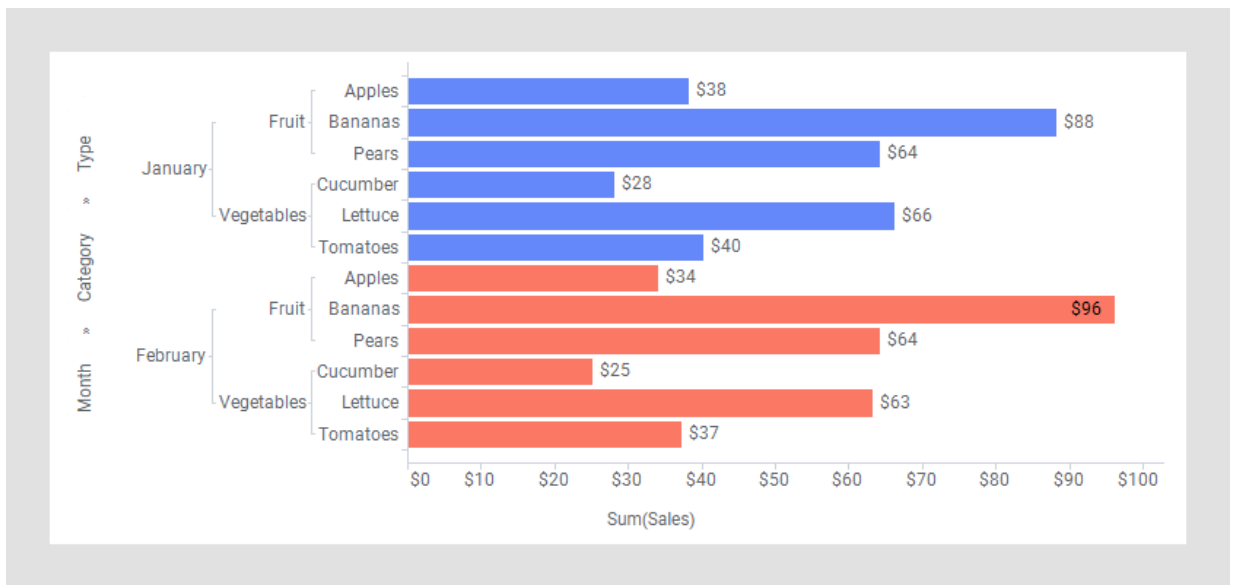
### Example



To apply the expressions used in this topic, it is assumed that [in-memory data](#) is used.

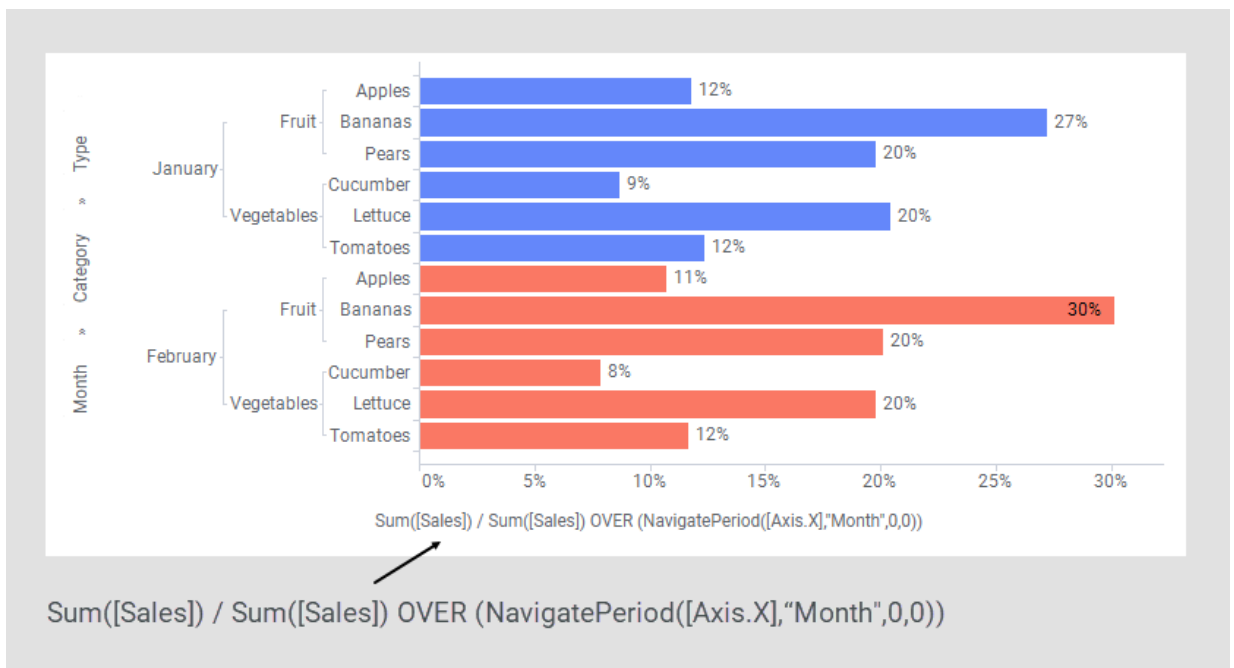
The bar chart shows sales figures in January and February for different types of fruits and vegetables, split also by their category. Assume you wish to display, expressed in percentage, each type's monthly contribution to the sales.





The Parent() method would work fine to return a type's contribution within the fruit and vegetables categories respectively, that is, one level up in the hierarchy, but it would not work that easily moving two steps up!

In this case the NavigatePeriod() method is a good choice. See the expression on the value axis below:



- The first argument, [Axis.X], refers to the hierarchy axis, in this case the Category axis. Note that the Category axis is always referred to as Axis.X, no matter if the orientation of the bar chart is horizontal or vertical.
- The second argument, "Month", navigates up to the Month level, that is, two steps up in the hierarchy. "Month" could just as well be replaced by 2 in the expression.
- The third argument, 0, specifies that no navigation takes place sideways in the hierarchy.
- The fourth argument, another 0, keeps the Month level.

For example, 27% of the total sales in January is sales of bananas ( $88 / (38 + 88 + 64 + 28 + 66 + 40)$ ).



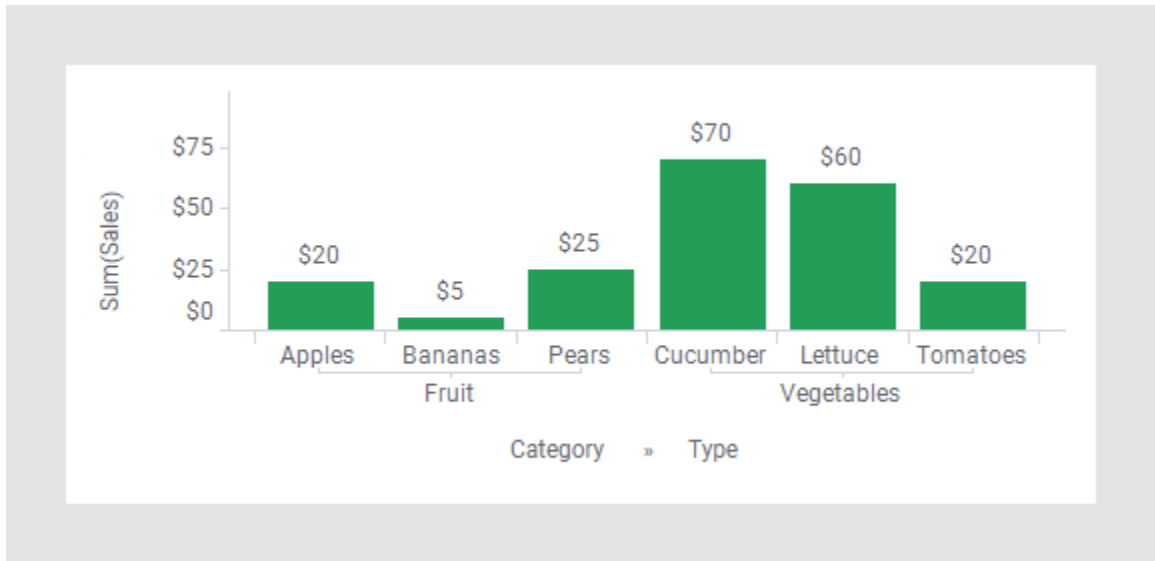
The percentages are rounded off.

### Comparing the NavigatePeriod() and Parent() methods

The NavigatePeriod() method can replace many of the other node navigation methods.

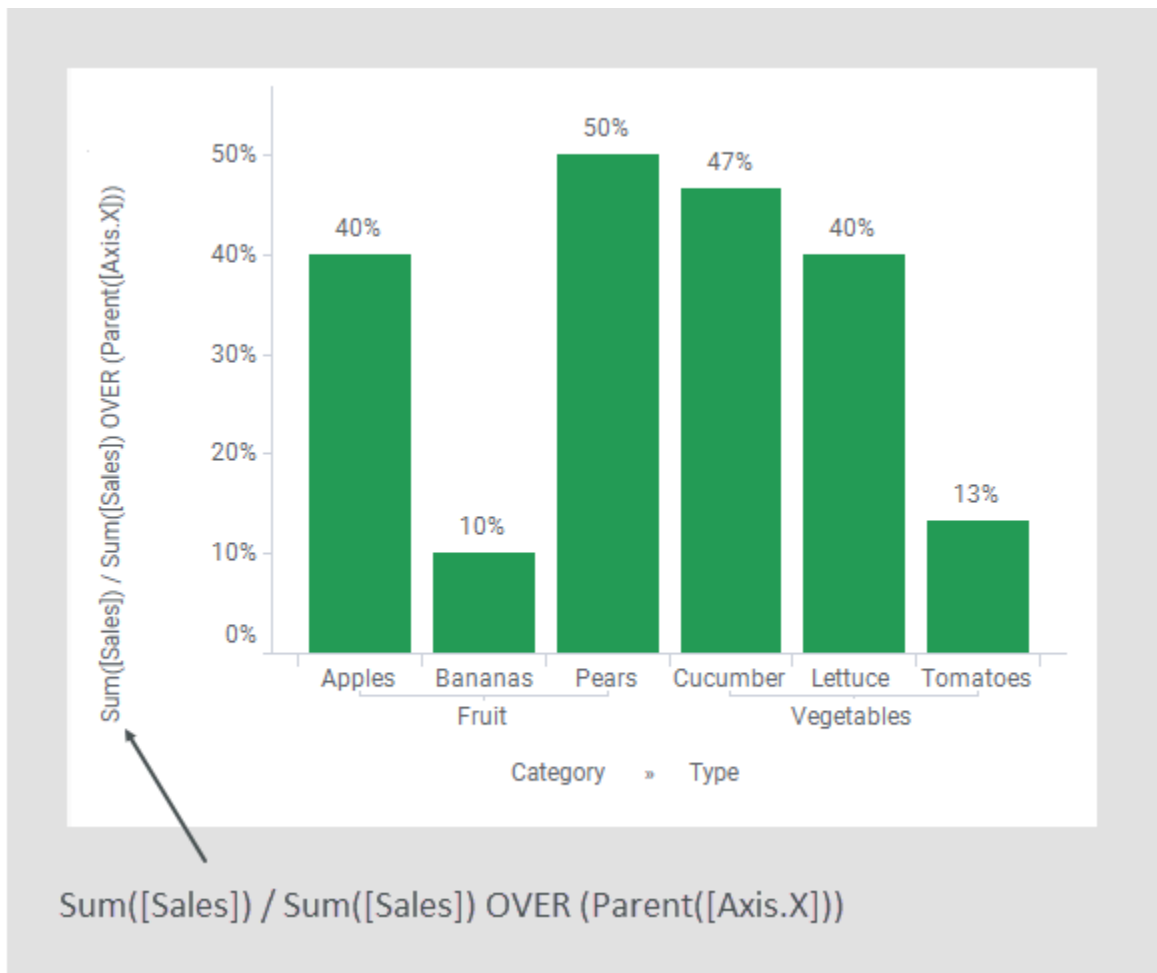
As an example, see the following comparison of the NavigatePeriod() and Parent() methods illustrated by means of the bar charts below.

The bar chart at the top shows the sum of sales for some fruits and vegetables.

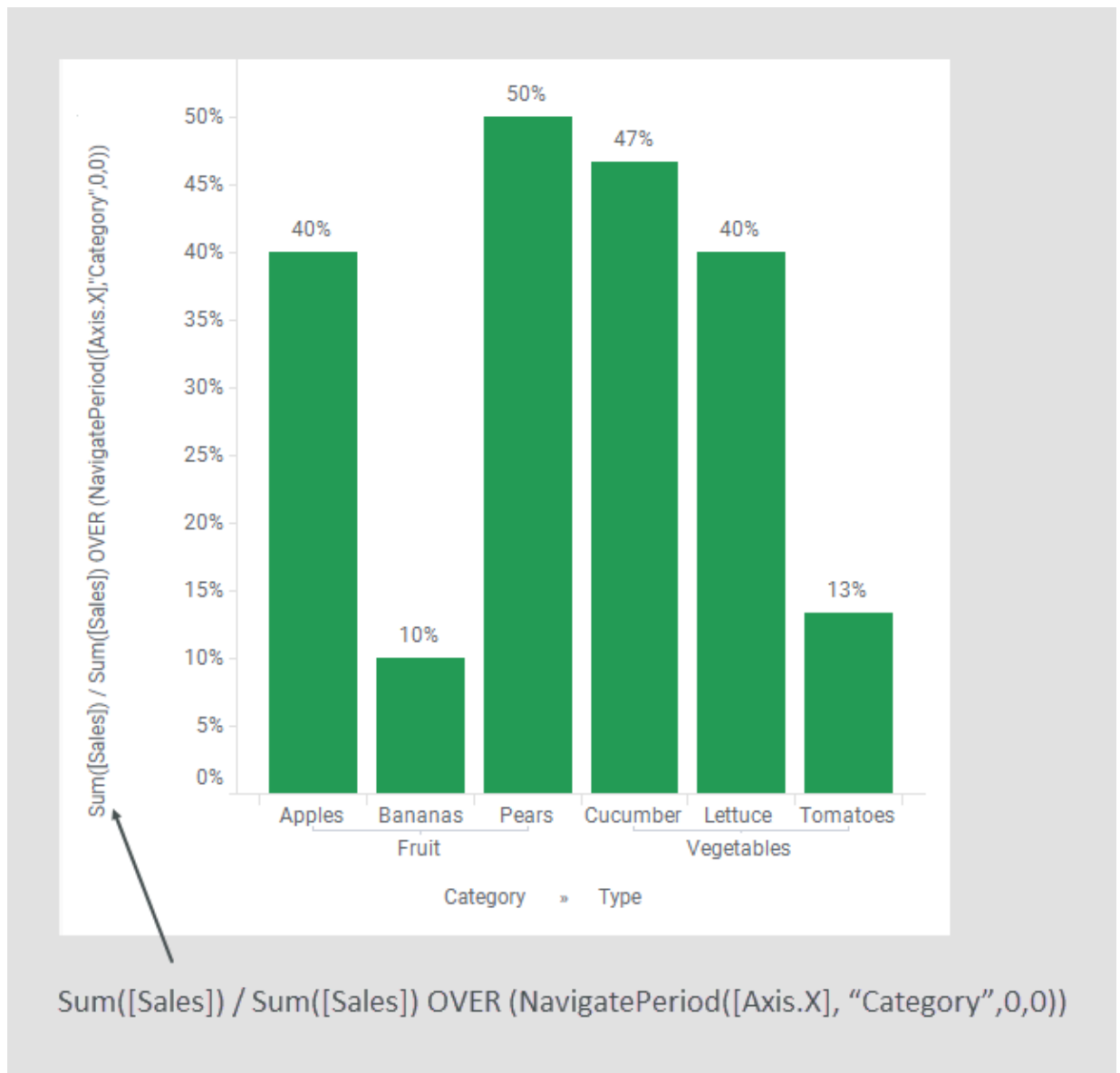


Let us focus on the Bananas. Assume you want to calculate the Bananas contribution to the total sales of fruit, expressed as a percentage. Both of the expressions in the bar charts below return identical results.

#### The Parent() method



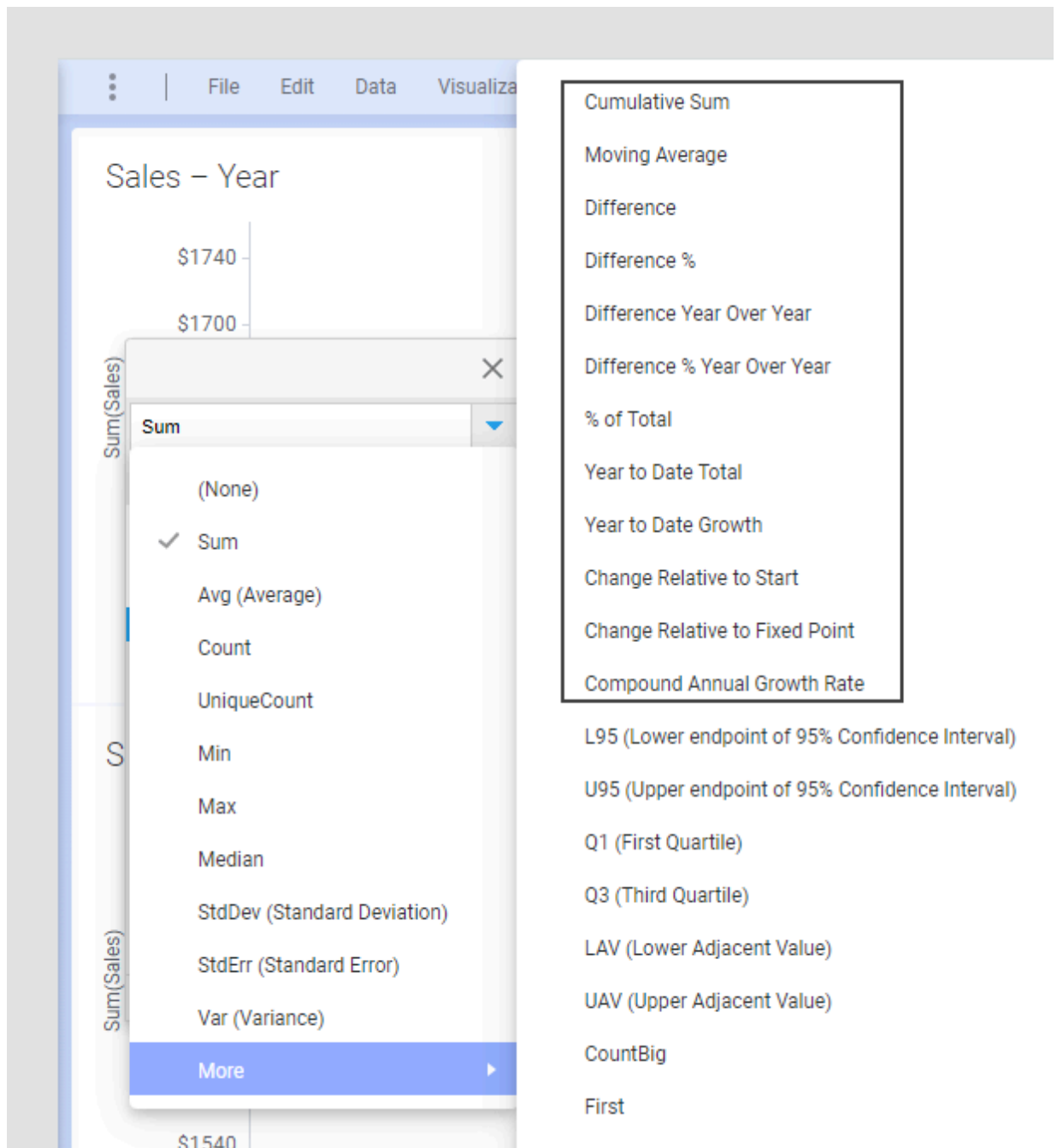
**The NavigatePeriod() method**



The "Category", 0, 0 part in the NavigatePeriod() expression corresponds to the Parent level, that is, it tells to navigate one level up to Category in the hierarchy, not to move sideways, and keep the level.

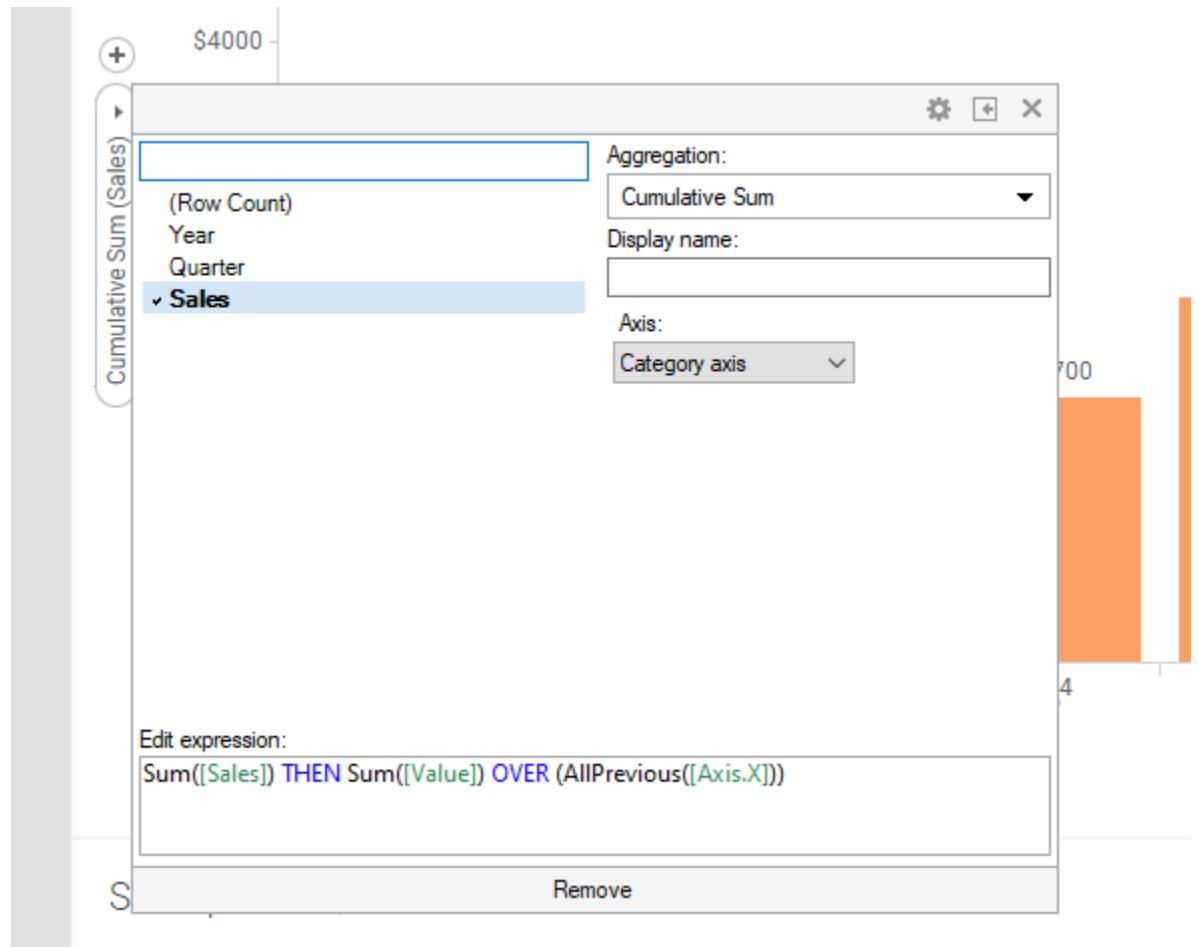
## Shortcuts to frequent custom expressions

Some custom expressions are more commonly used than others, for example, moving average and cumulative sum. Shortcuts to a number of the most common expressions are provided in the list of aggregation methods in the column selector on an axis, so you do not need to type these expressions yourself. Which shortcuts that are available depends on what kind of data you have selected on the axis in question in the visualization.

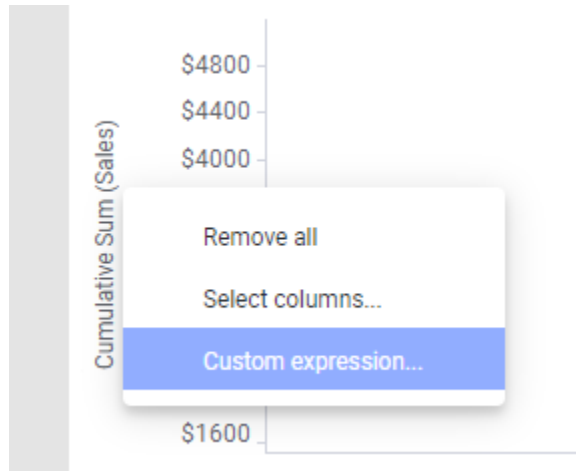


The entire expressions for the shortcuts as well as the standard aggregation methods can be viewed and edited.

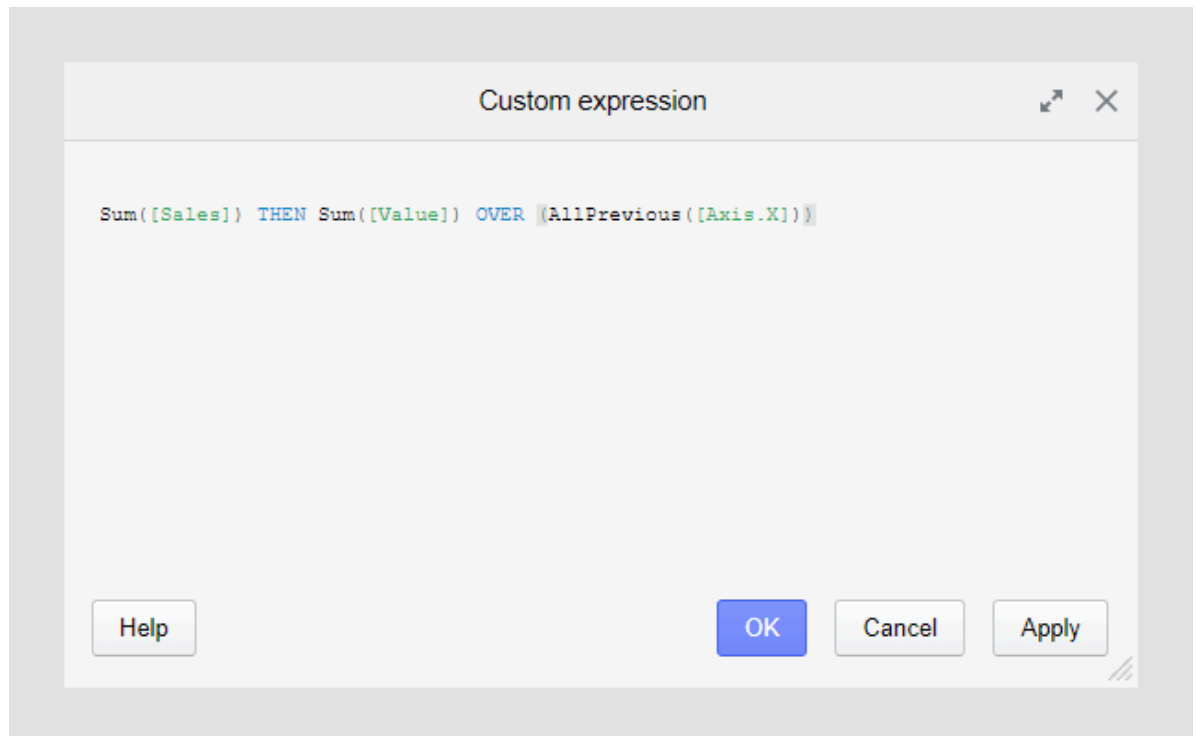
- In the installed client, this is done in the expanded axis selector:



- In the web client, right-click the column selector, and select **Custom expression**:



The Custom expression dialog will open, where you can view and edit the expression.



## The expressions

When you view the expressions for the shortcuts in the Custom expression dialog, you notice that they contain (as you can see in the image above)

- an **OVER** statement and a [node navigation](#) method
- a **THEN** keyword and a **[Value]** column.

To understand how to evaluate the expressions, these concepts need an explanation.

### The OVER statement and the node navigation method

The OVER statement is a means to reference different slices of data when using various node navigation methods. When calculating the new value of a certain slice for the method expression in question,

- the OVER statement tells that slices outside of the current slice should be used when evaluating the expression,
- and the node navigation method tells which slices to use.

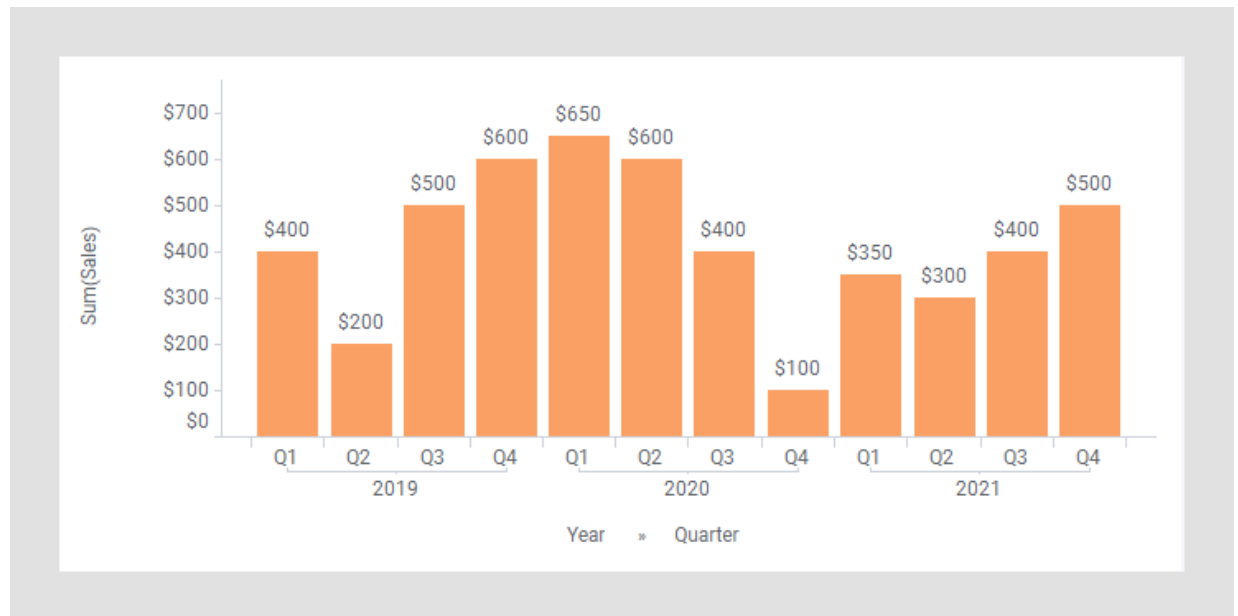
### The THEN keyword and the [Value] column

The use of the THEN keyword makes it possible to evaluate an expression on already aggregated data. The part of the expression before the THEN keyword is calculated first.

The temporary column called [Value] represents the result of the entire (already aggregated) expression before the THEN keyword.

### Examples of expression shortcuts

What is calculated when applying each of the shortcut expressions in the rectangle above is explained in the following topics. All the topics (except the Top Category topic) use the bar chart below as input.



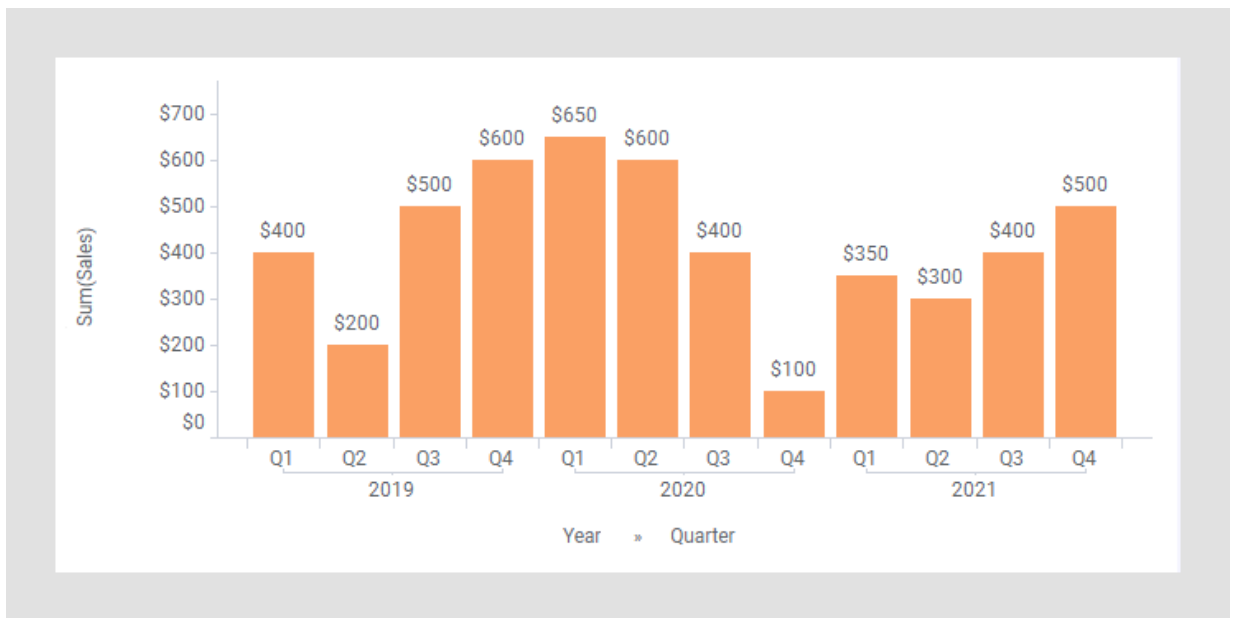
### Cumulative Sum

A Cumulative Sum, or running total, displays the total sum of data as it grows with time, or with any other series of data. The total contribution so far of a given measure is calculated for each item (slice) in the visualization.

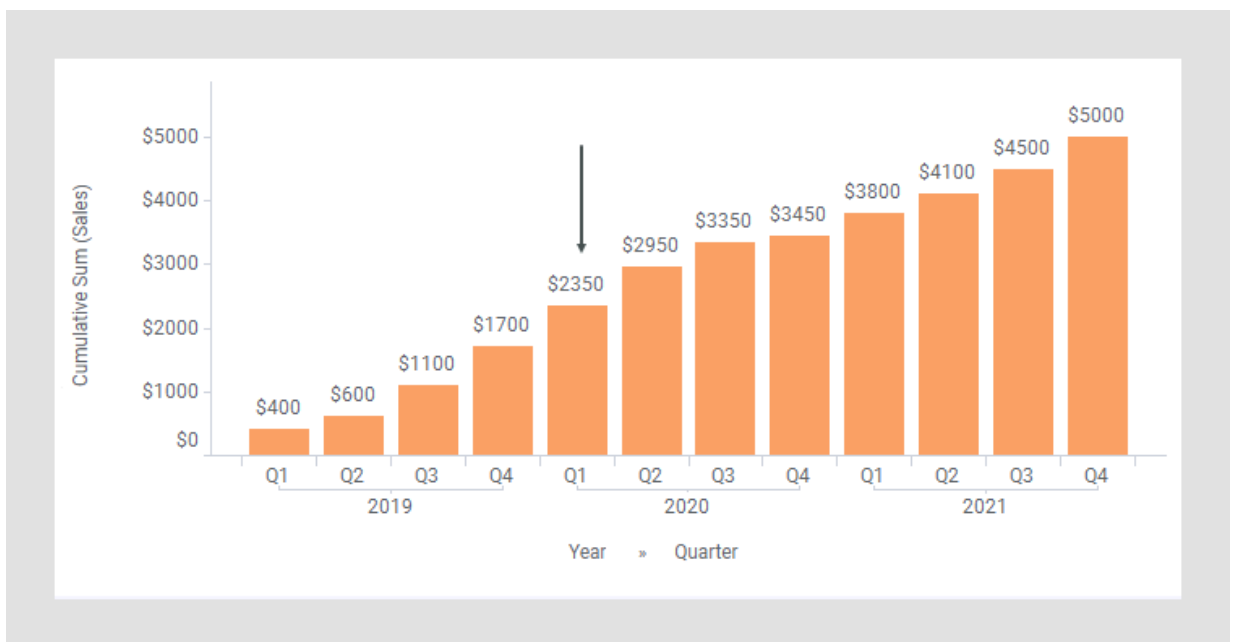
### Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.





In the next bar chart, the shortcut expression Cumulative Sum has been applied on the Value axis. (The arrow points to a bar used to explain what is calculated in the expression further down.)



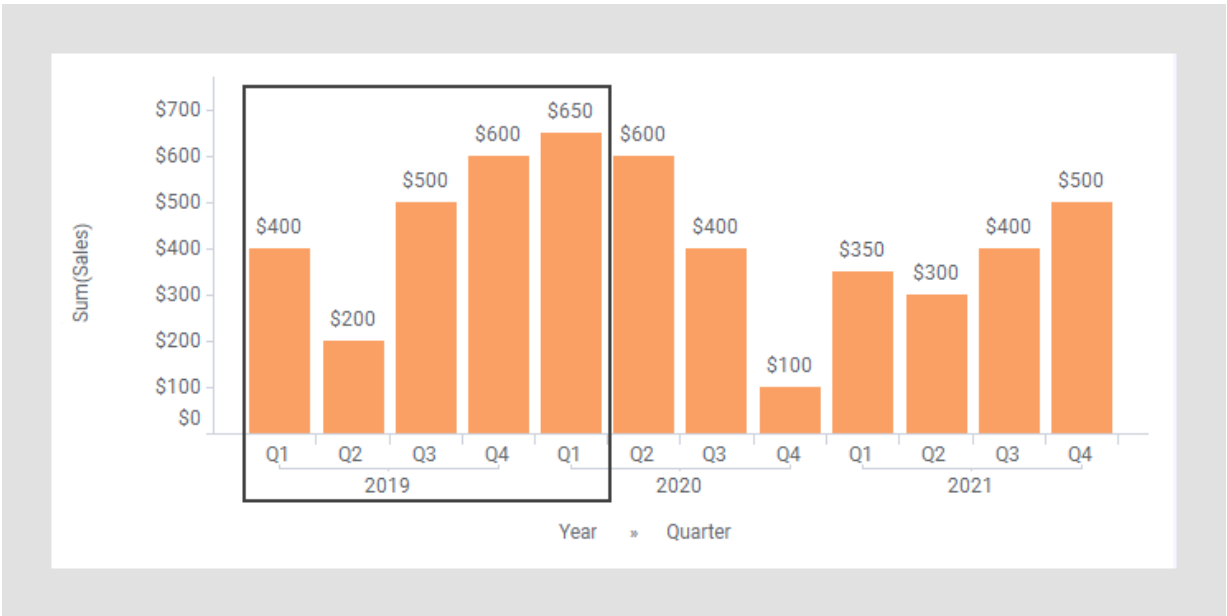
The following expression is used on the Value axis:

```
Sum([Sales]) THEN Sum([Value]) OVER (AllPrevious([Axis.X]))
```

The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following THEN. For each node in the visualization (in this case each bar), the last part of the expression sums the current node and all previous nodes.

For example, calculation of the Cumulative Sum for Q1, 2020 (see the arrow in the previous image):

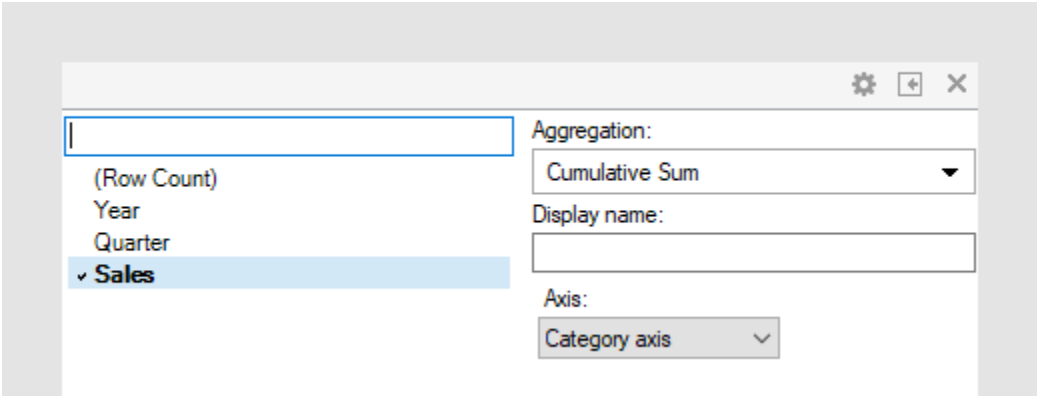
$\$2350 = \$400 + \$200 + \$500 + \$600 + \$650$  (the sum of the bars in the rectangle below).



See also [AllPrevious\(\)](#).

**The column selector in expanded mode**

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



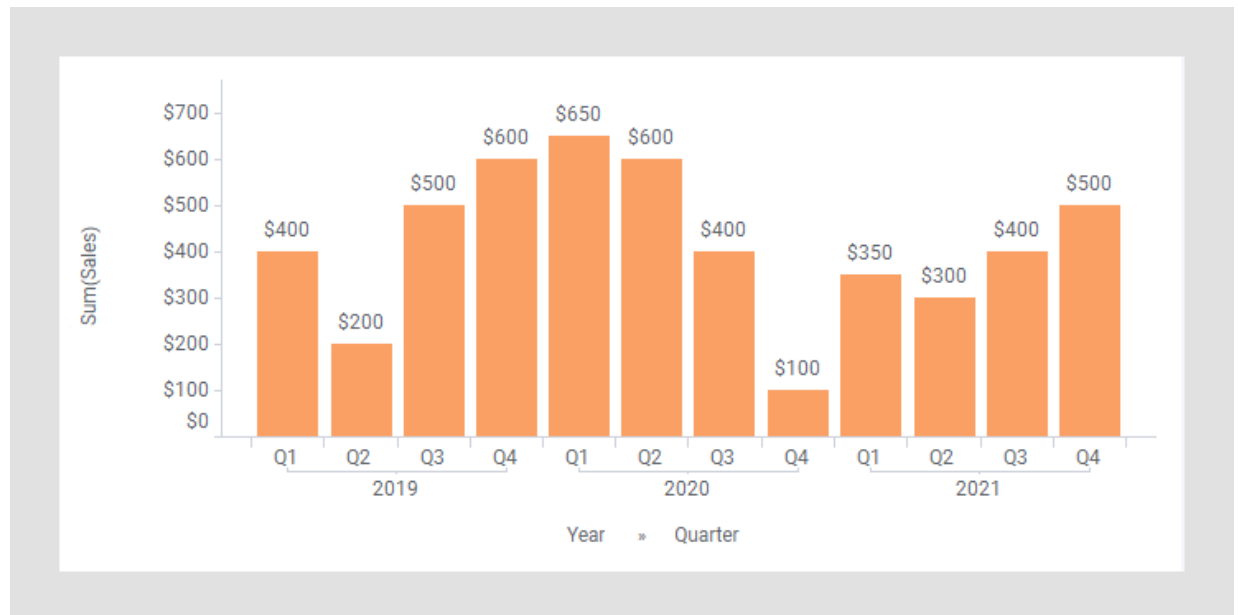
Display name	Optionally, specify a different display name instead of the default "Cumulative Sum ([Column Name])".
Axis	<p>Specify the axis over which to calculate the nodes.</p> <p>Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.</p>

**Moving Average**

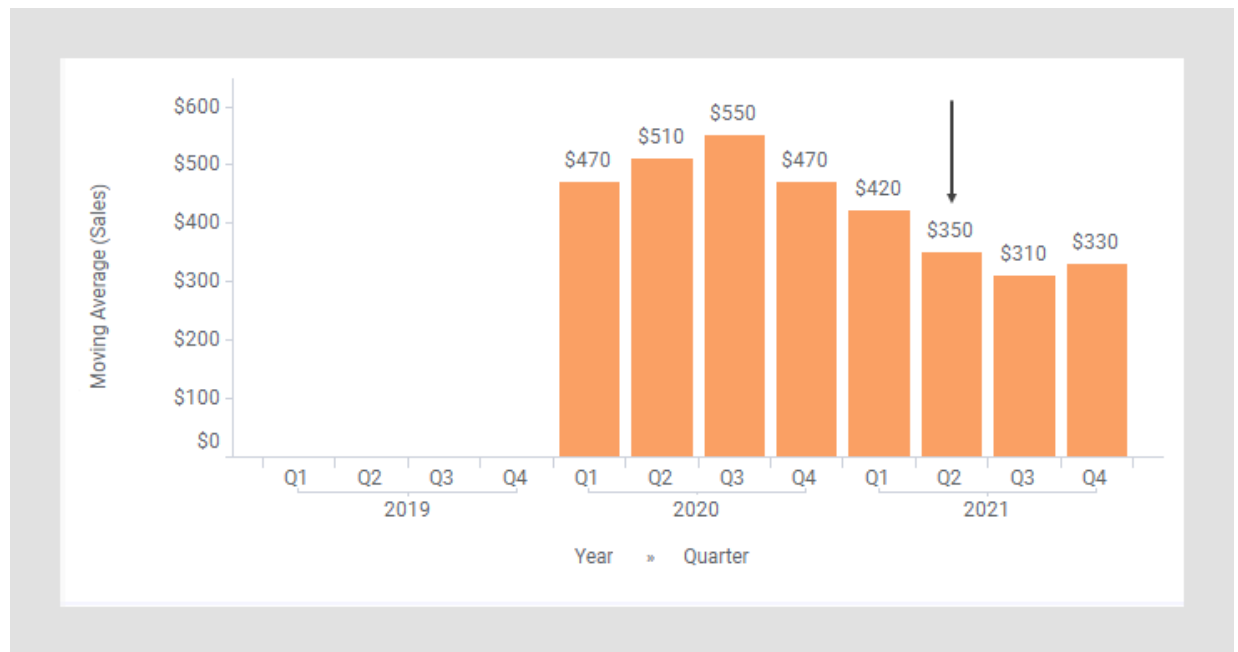
A Moving Average, also called rolling average, rolling mean, or running average, calculates the average of the items (slices) within a specified interval. For example, if the interval size is set to three, the average is calculated on the current item and the two preceding items. The purpose of using a moving average is often to smooth out short-term fluctuations and to spot long-term trends.

## Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression Moving Average has been applied on the Value axis. (The arrow points to a bar used to explain what is calculated in the expression further down.)



The following expression is used on the Value axis (here the interval is set to 5).

```
Sum([Sales]) THEN Avg([Value]) OVER (LastPeriods(5,[Axis.X])) THEN If(Count() OVER (LastPeriods(5,[Axis.X]))=5,[Value],null)
```

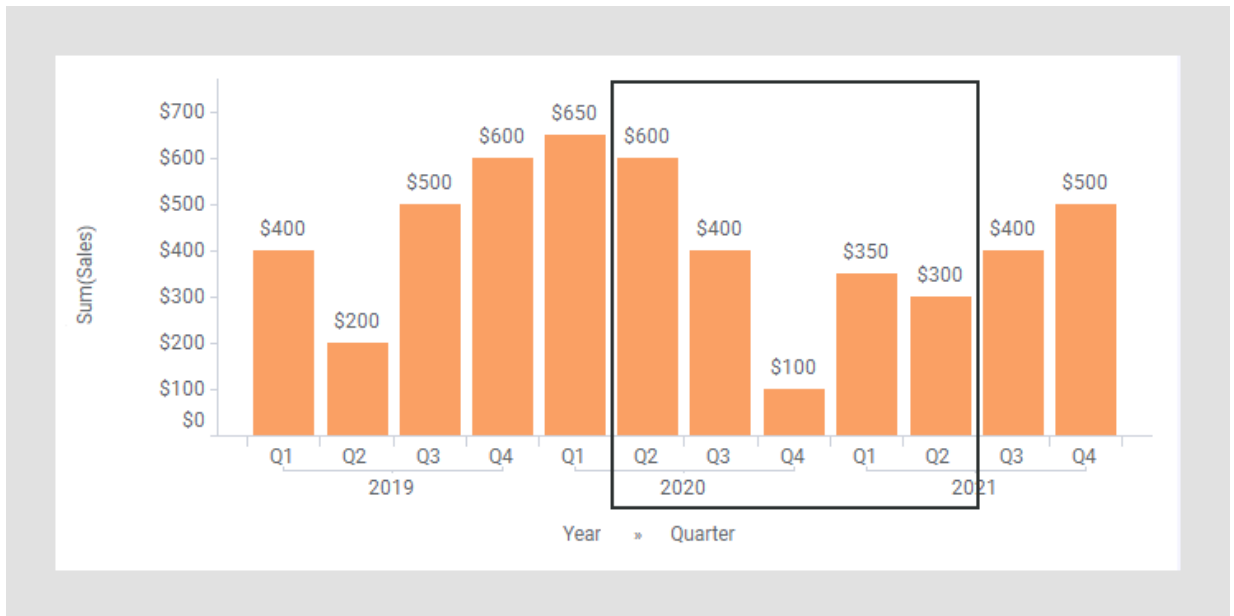
The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following the first THEN.

The middle part of the expression calculates, for each node in the visualization (in this case each bar), the average of the node and the four previous nodes. The final part, after the second THEN, ensures that no moving average is shown if there are less than 5 bars.

For example, calculation of the Moving Average for Q2, 2021 (see the arrow in the previous image):

$$\$350 = (\$600 + \$400 + \$100 + \$350 + \$300) / 5$$

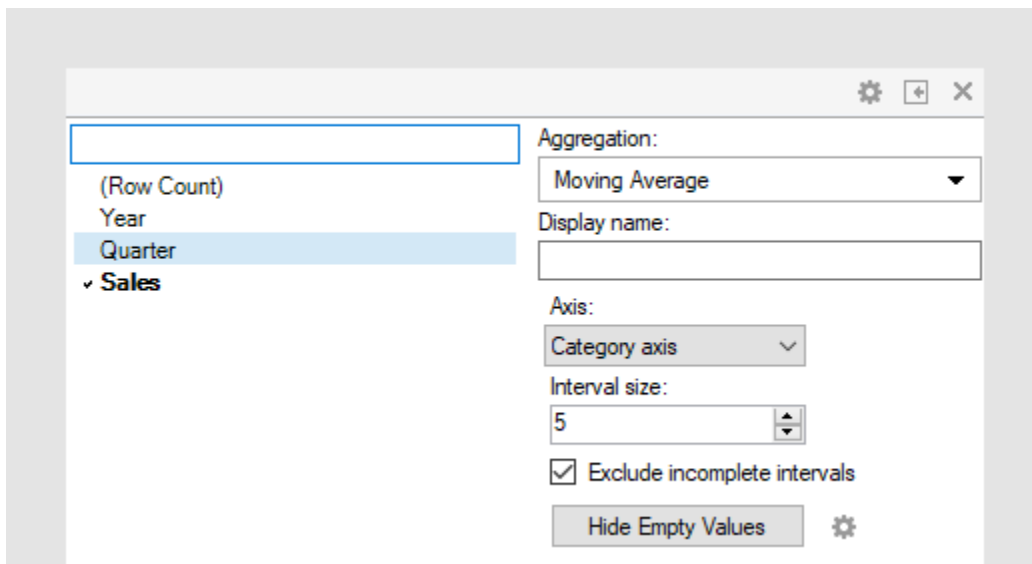
(the sum of the node and the four previous nodes within the rectangle below divided by 5)




See also [LastPeriods\(\)](#).

### The column selector in expanded mode

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



Display name	Optionally, specify a different display name instead of the default "Moving Average ([Column Name])".
--------------	---

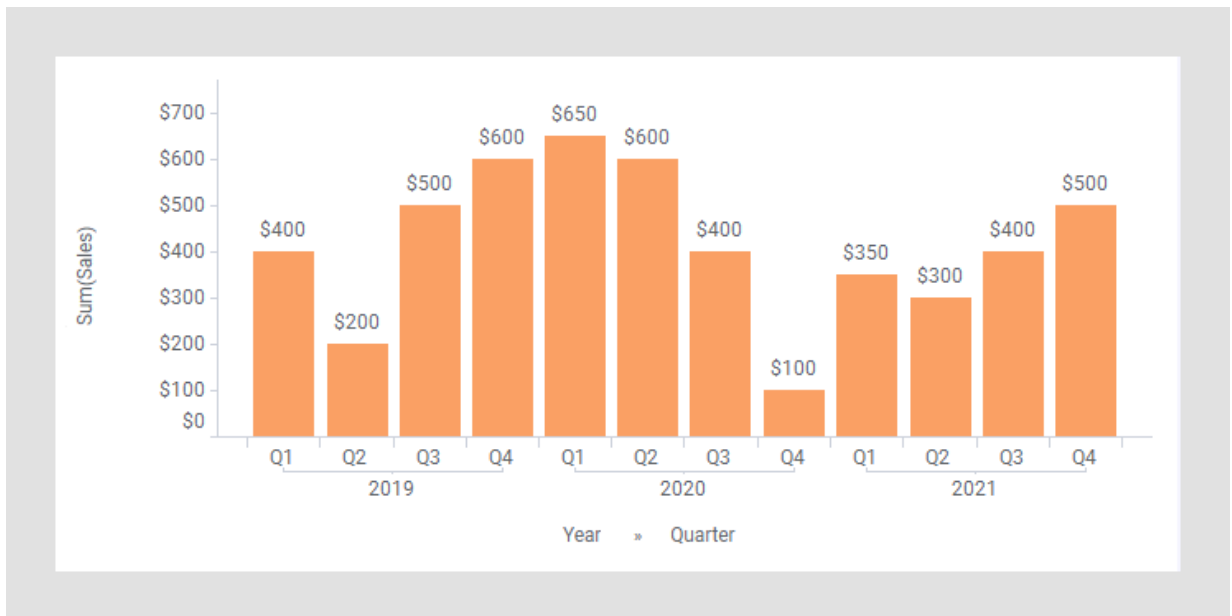
Axis	Specify the axis over which to calculate the nodes.  Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.
Interval size	Define the size of the window to calculate the average within.
Exclude incomplete intervals	Determine whether to calculate averages for intervals that lack values for one or more of the nodes in the interval.  If the check box is cleared, then the average value will be calculated using those values that are available only. If the check box is selected, nodes within incomplete intervals will be left empty.
Hide Empty Values	Creates a Show/Hide rule that hides all empty values. Click the properties symbol  to access the Show/Hide properties to edit or remove the rule.

## Difference

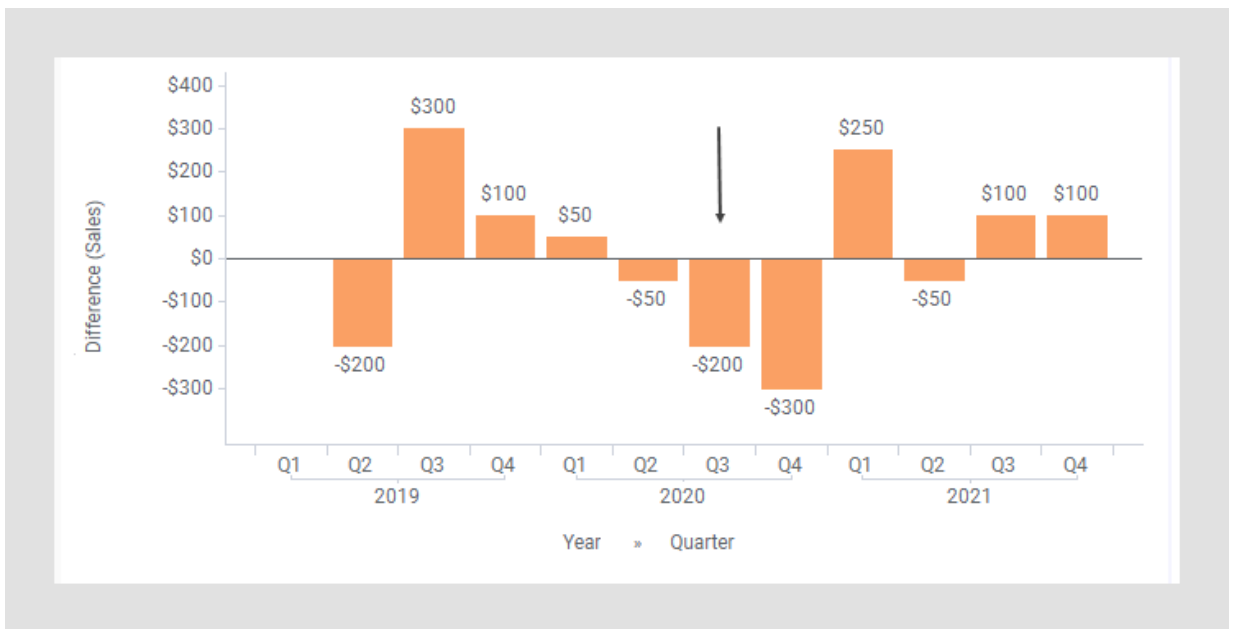
The Difference expression shortcut calculates the difference between two nodes. The nodes should be on the same level in the hierarchy, and you can specify how many steps there should be between the nodes to be compared.

## Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression Difference has been applied on the Value axis. (The arrow points to a bar used to explain what is calculated in the expression further down.)



The following expression is used on the Value axis.

```
Sum([Sales]) THEN [Value]- First([Value]) OVER (NavigatePeriod([Axis.X],0,-1))
```

The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following THEN.

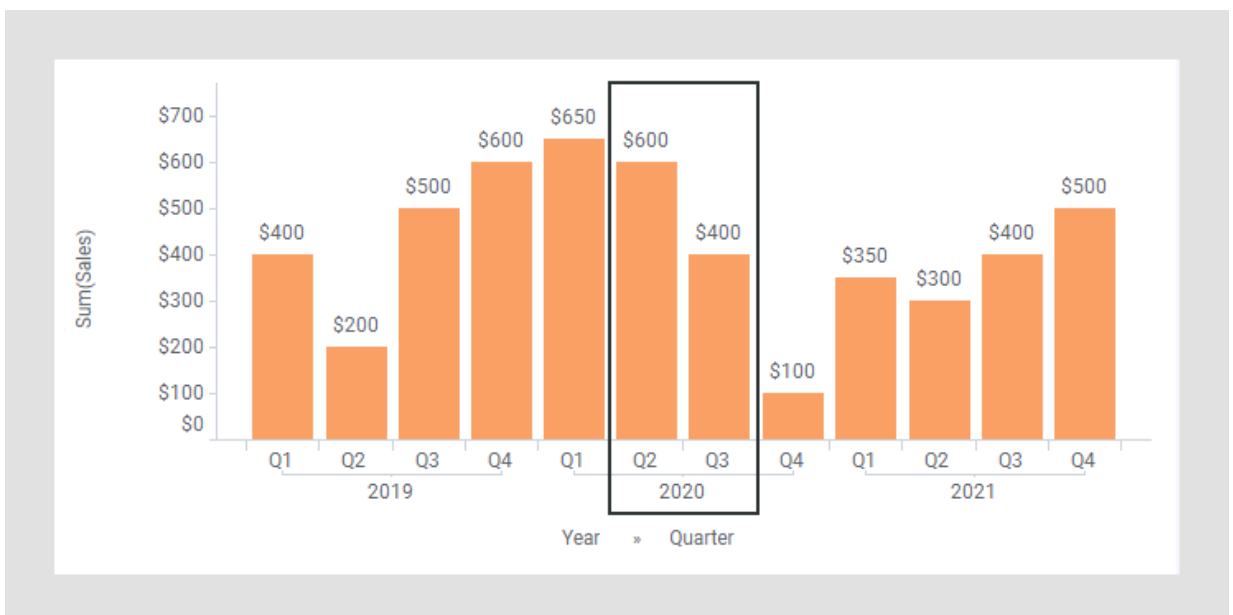
In this latter part of the expression, each node gets subtracted by its preceding node (because the number of steps is set to -1). See [NavigatePeriod\(\)](#) for details.

The aggregation First() is just used because an aggregation is required before the OVER keyword. It will take you to the first value of the node to subtract, but since each node only has one value, this corresponds to picking the value of the node.

For example, calculation of the Difference for Q3, 2020 (see the arrow in the previous image):

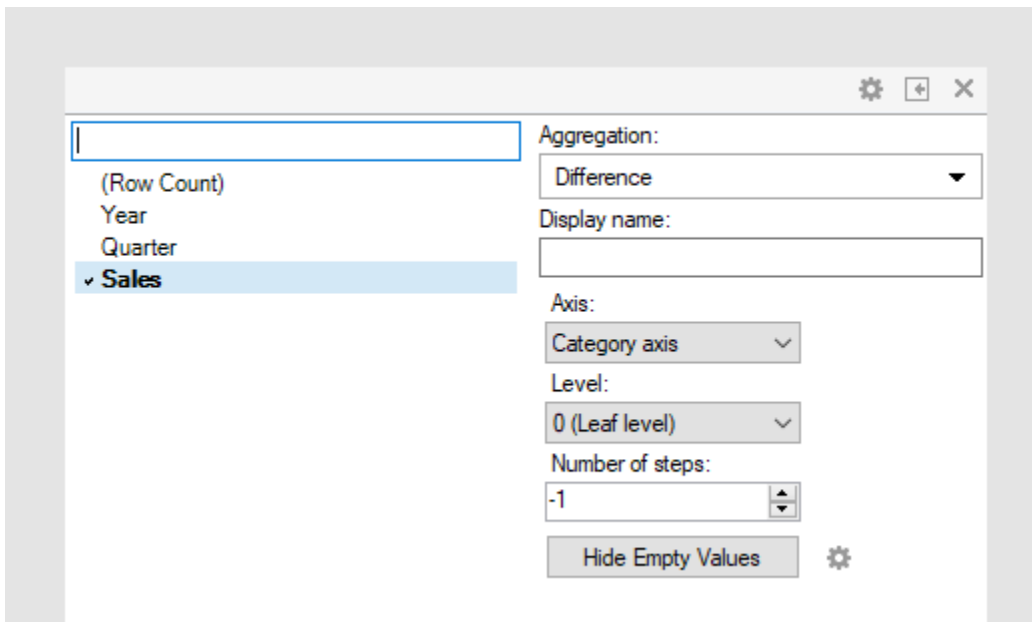
$-\$200 = \$400 - \$600$


(the difference of the node and its previous node, see the rectangle).



## The column selector in expanded mode

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



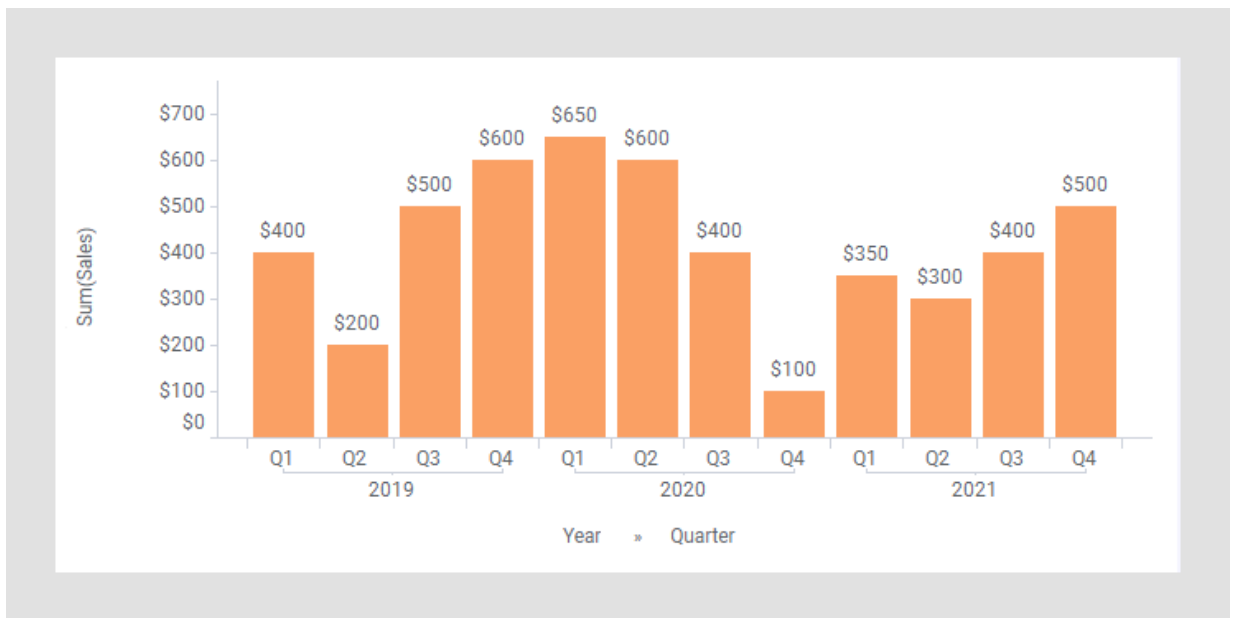
Display name	Optionally, specify a different display name instead of the default "Difference ([Column Name])".
Axis	Specify the axis over which to calculate the nodes.  Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.
Level	Define the hierarchy level to compare values at. You can either specify a named level such as "Year", "Quarter" or "Month", or you can set a number, which moves you up or down in the currently set hierarchy on the selected axis.  If the level is 0, you will always compare the current leaf level, even if you change it in the visualization, for example, using a hierarchy slider.  If the level is set to "Year", then you are in fact doing a <a href="#">Difference Year over Year</a> calculation, and the aggregation selector will change correspondingly.
Number of steps	Define the number of steps between the nodes to compare. If the number of steps is -1, then the difference will be calculated between the current node and the one before it. If the number of steps is 1, then the difference will be calculated for the current node and the one following it.
Hide Empty Values	Creates a Show/Hide rule that hides all empty values. Click the properties symbol  to access the Show/Hide properties to edit or remove the rule.

## Difference %

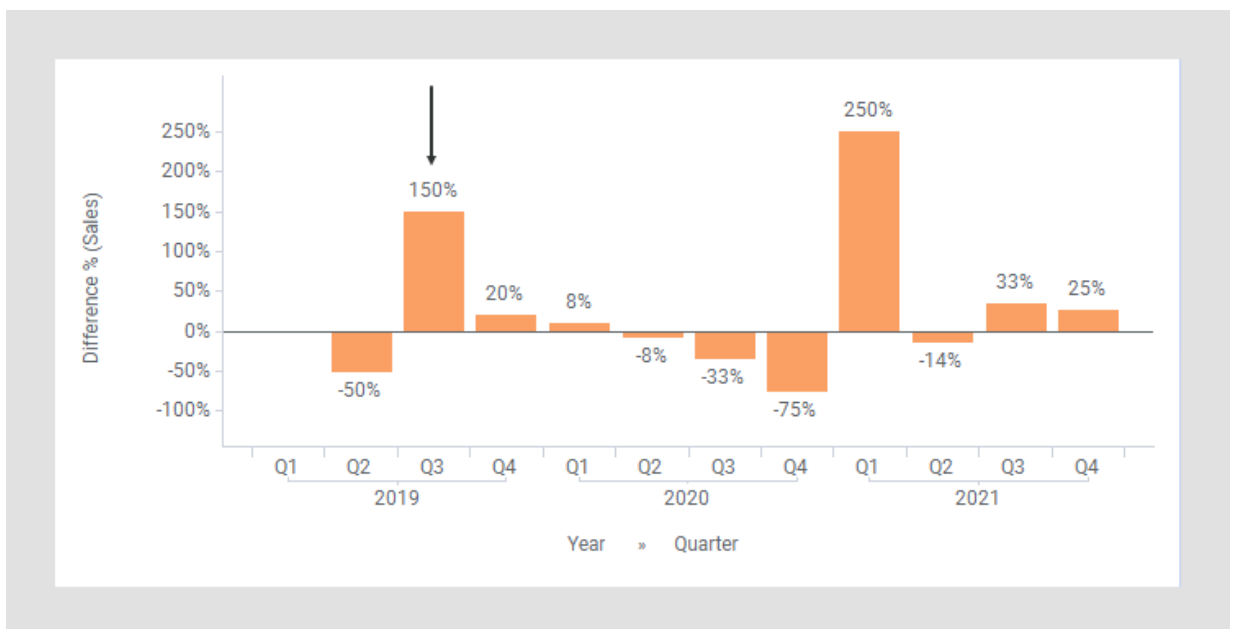
The Difference % expression shortcut calculates the difference in percent between two nodes. The nodes should be on the same level in the hierarchy, and you can specify how many steps there should be between the nodes to be compared.

## Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression Difference % has been applied on the Value axis. (The arrow points to a bar used to explain what is calculated in the expression further down.)



The following expression is used on the Value axis.

```
Sum([Sales]) THEN ([Value]/ First([Value]) OVER (NavigatePeriod([Axis.X],0,-1)))
```

The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following the first THEN.

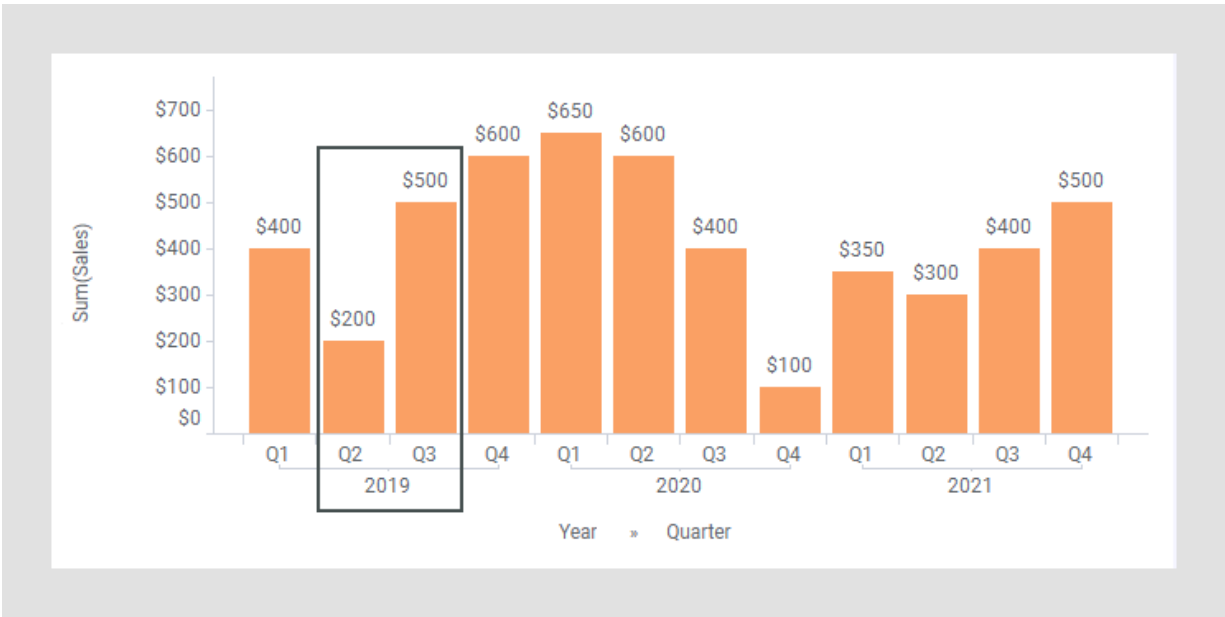
In this latter part of the expression, each node gets divided by the preceding node (because the number of steps is set to -1, see [NavigatePeriod\(\)](#) for details), and then this result is subtracted by 1 to get the percentage.

The aggregation First() is just used because an aggregation is required before the OVER keyword. It will take you to the first value of the node to subtract, but since each node only has one value, this corresponds to picking the value of the node.

For example, calculation of the Difference % for Q3, 2019 (see the arrow in the previous image):

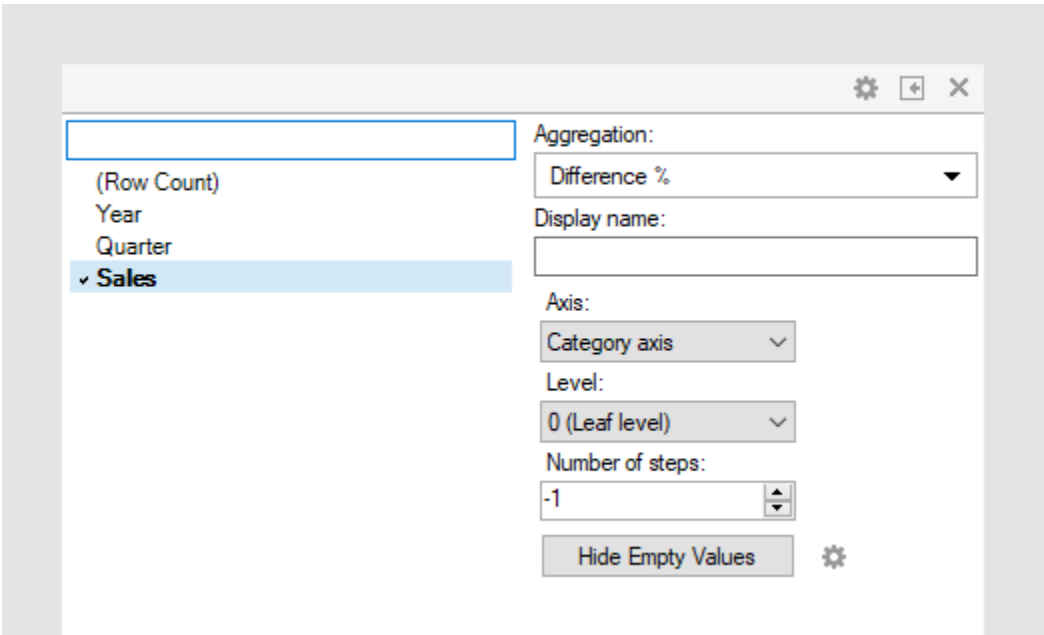


150%=\$500 / \$200 - 1  
(see the rectangle below).




**The column selector in expanded mode**

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



Display name	Optionally, specify a different display name instead of the default "Difference % ([Column Name])".
Axis	<p>Specify the axis over which to calculate the nodes.</p> <p>Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.</p>

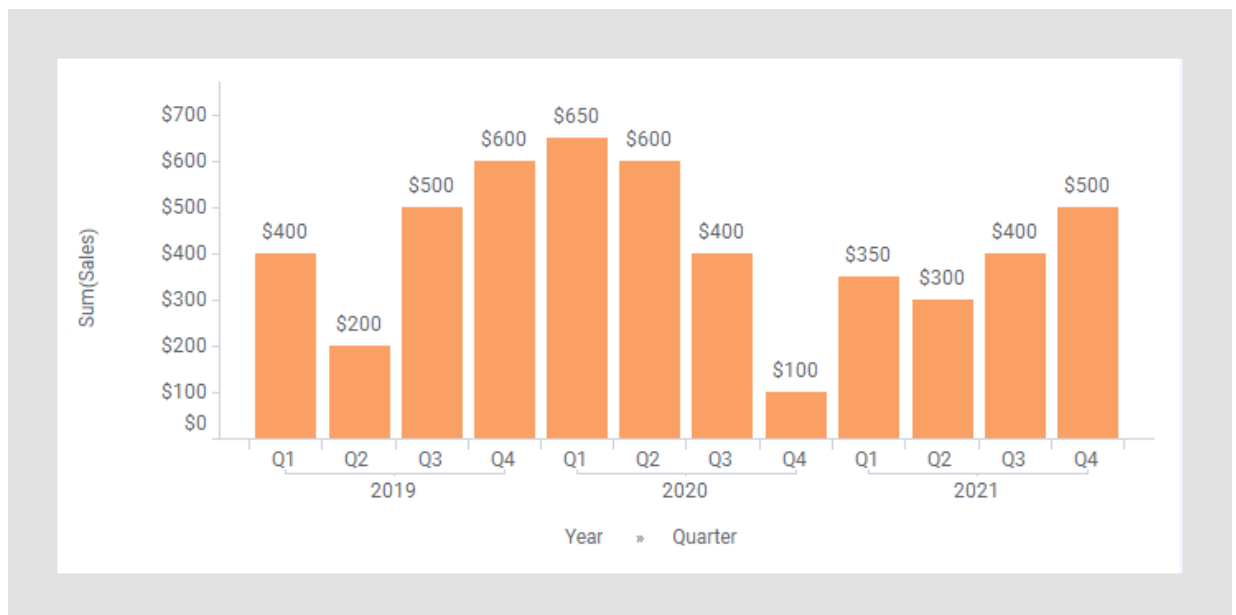
Level	<p>Define the hierarchy level to compare values at. You can either specify a named level such as "Year", "Quarter" or "Month", or you can set a number, which moves you up or down in the currently set hierarchy on the selected axis.</p> <p>If the level is 0, you will always compare the current leaf level, even if you change it in the visualization, for example, using a hierarchy slider.</p> <p>If the level is set to "Year", then you are in fact doing a <a href="#">Difference % Year over Year</a> calculation, and the aggregation selector will change correspondingly.</p>
Number of steps	<p>Define the number of steps between the nodes to compare. If the number of steps is -1, then the difference will be calculated between the current node and the one before it. If the number of steps is 1 then the difference will be calculated for the current node and the one following it.</p>
Hide Empty Values	<p>Creates a Show/Hide rule that hides all empty values. Click the properties symbol  to access the Show/Hide properties to edit or remove the rule.</p>

## Difference Year Over Year

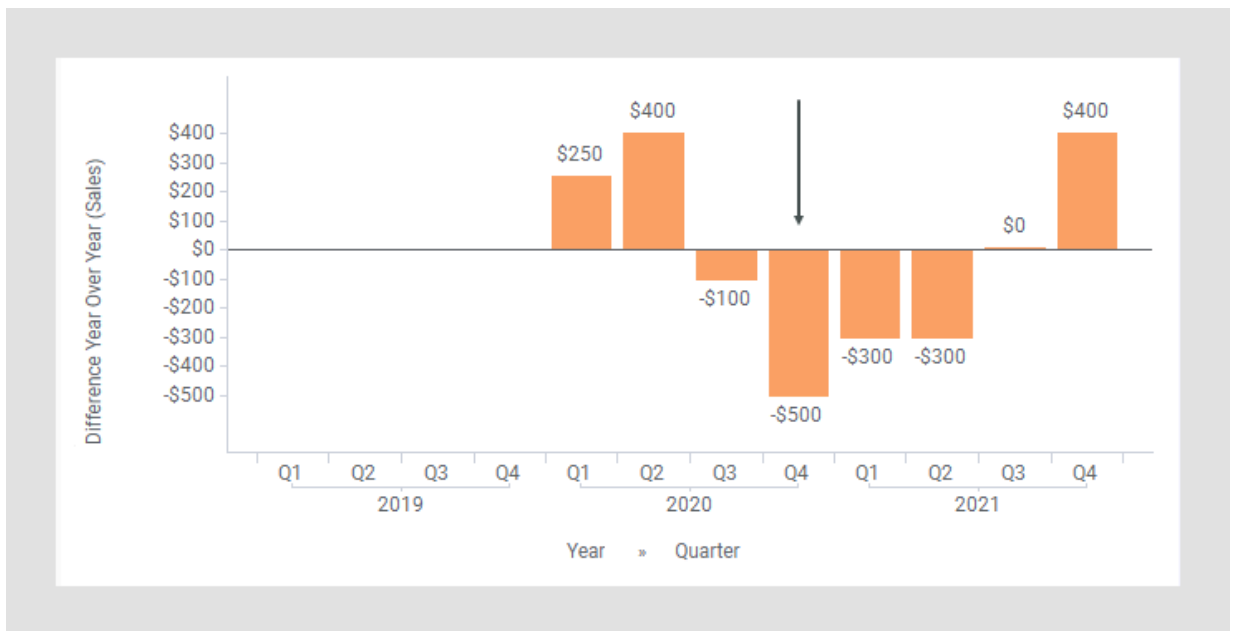
The Difference Year Over Year expression shortcut calculates the difference between corresponding nodes in different years. For example, you can compare the sales in the first quarter of a given year with the first quarter the year before, or with the first quarter two years ago.

### Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression Difference Year Over Year has been applied on the Value axis. (The arrow points to a bar used to explain what is calculated in the expression further down.)



The following expression is used on the Value axis:

```
Sum([Sales]) THEN [Value]- First([Value]) OVER (NavigatePeriod([Axis.X],"Year",-1))
```

The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following THEN.

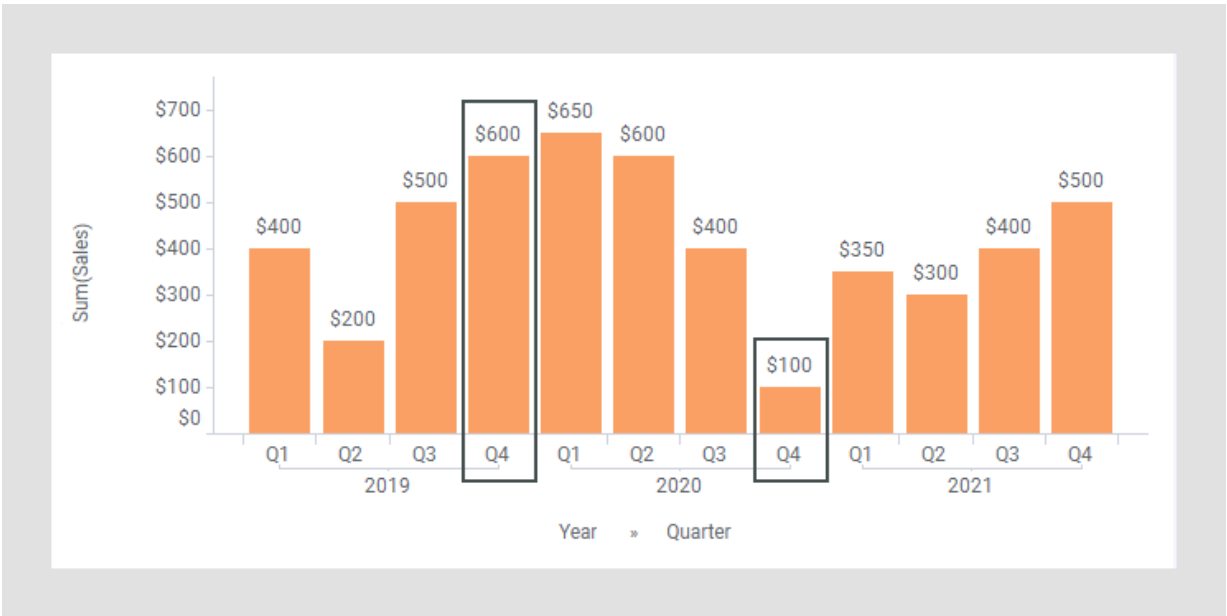
In this latter part of the expression, each node gets subtracted by the corresponding node the year before, because the number of steps is set to -1. If -1 is changed to -2, each node gets subtracted by the corresponding node two years ago. See [NavigatePeriod\(\)](#) for details.

The aggregation First() is just used because an aggregation is required before the OVER keyword. It will take you to the first value of the node to subtract, but since each node only has one value, this corresponds to picking the value of the node.

For example, calculation of the Difference Year Over Year for Q4, 2020 (see the arrow in the previous image):

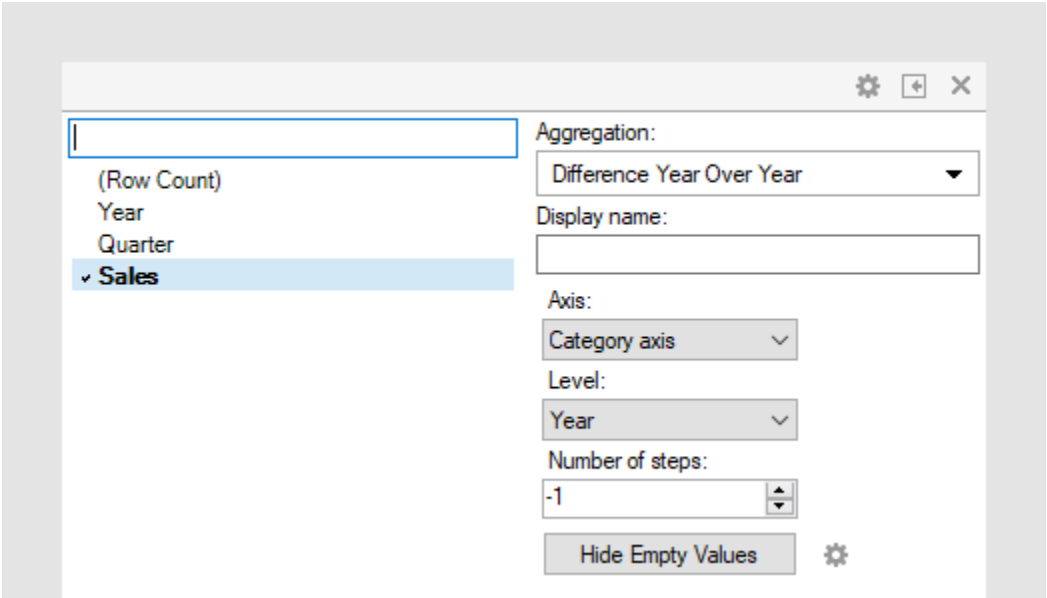
$$-\$500 = \$100 - \$600$$

(see the rectangles below)




**The column selector in expanded mode**

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



Display name	Optionally, specify a different display name instead of the default "Difference Year Over Year ([Column Name])".
Axis	<p>Specify the axis over which to calculate the nodes.</p> <p>Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.</p>

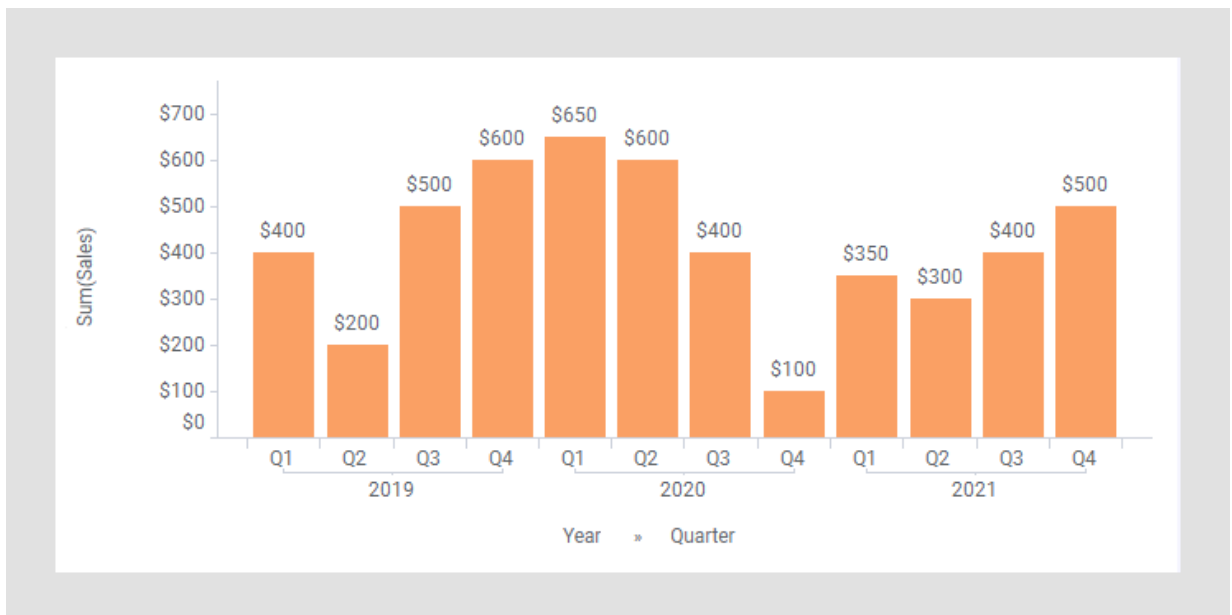
Level	<p>Define the hierarchy level to compare values at. You can either specify a named level such as "Year", "Quarter" or "Month", or you can set a number, which moves you up or down in the currently set hierarchy on the selected axis.</p> <p>If the level is 0 you will always compare the current leaf level, even if you change it in the visualization, for example, using a hierarchy slider.</p> <p>If the level is set to something other than "Year", then you are in fact doing a regular <a href="#">Difference</a> calculation, and the aggregation selector will change correspondingly.</p>
Number of steps	<p>Define the number of steps between the nodes to compare. If the number of steps is -1, then the difference will be calculated between the current node and the one before it. If the number of steps is 1, then the difference will be calculated for the current node and the one following it.</p>
Hide Empty Values	<p>Creates a Show/Hide rule that hides all empty values. Click the properties symbol  to access the Show/Hide properties to edit or remove the rule.</p>

## Difference % Year Over Year

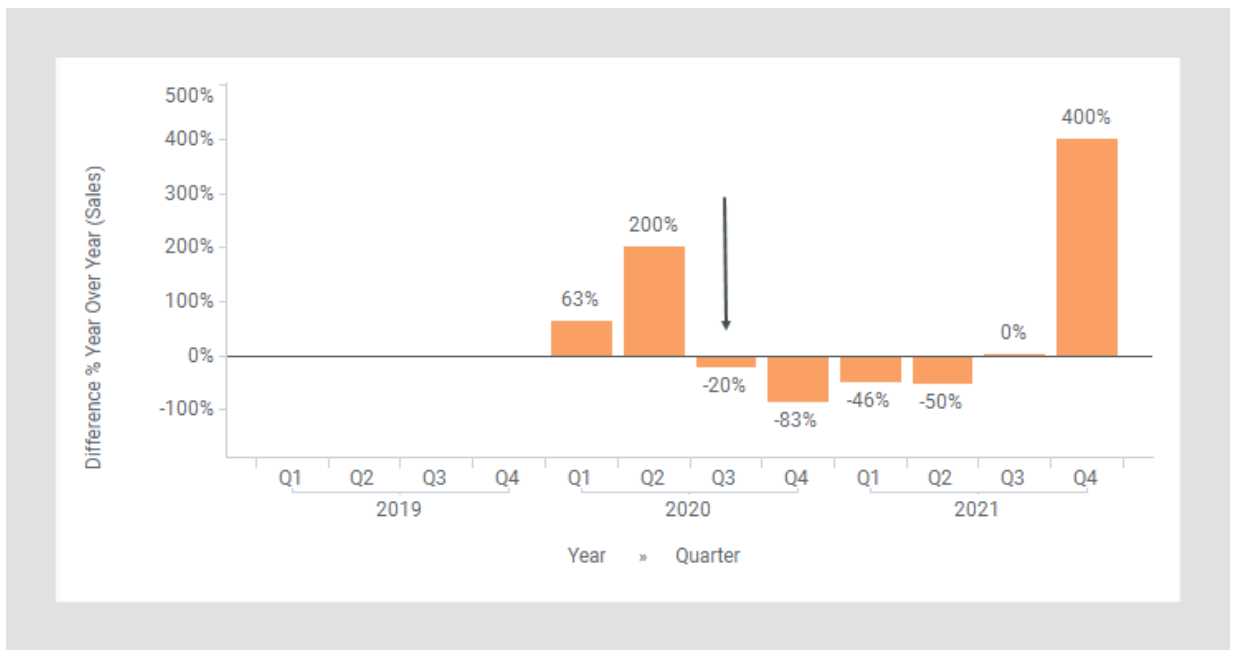
The Difference % Year Over Year expression shortcut calculates the percentage difference between corresponding nodes in different years. For example, you can compare the sales in the first quarter of a given year with the first quarter the year before, or with the first quarter two years ago.

### Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression Difference % Year Over Year has been applied on the Value axis. (The arrow points to a bar used to explain what is calculated in the expression further down.)



The following expression is used on the Value axis.

```
Sum([Sales]) THEN ([Value]/ First([Value]) OVER
(NavigatePeriod([Axis.X], "Year", -1))) - 1
```

The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following THEN.

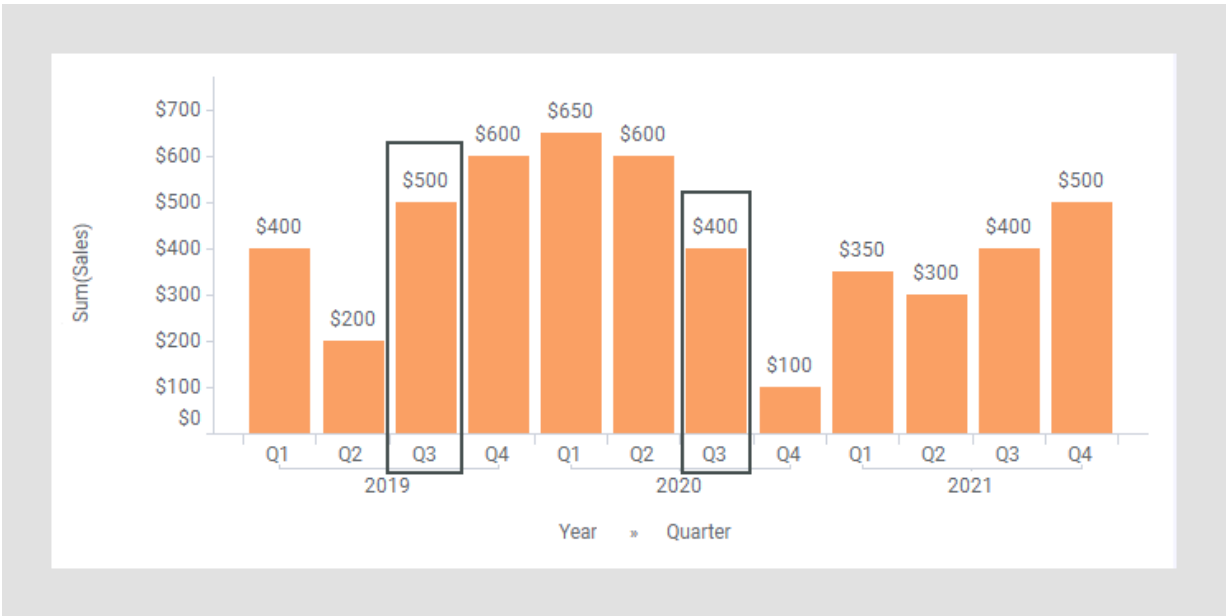
In this latter part of the expression, each node gets divided by the corresponding node one year ago (because the number of steps is set to -1, see [NavigatePeriod\(\)](#) for details), and then this result is subtracted by 1 to get the percentage.

The aggregation First() is just used because an aggregation is required before the OVER keyword. It will take you to the first value of the node to subtract, but since each node only has one value, this corresponds to picking the value of the node.

For example, calculation of the Difference % Year Over Year for Q3, 2020 (see the arrow in the previous image):

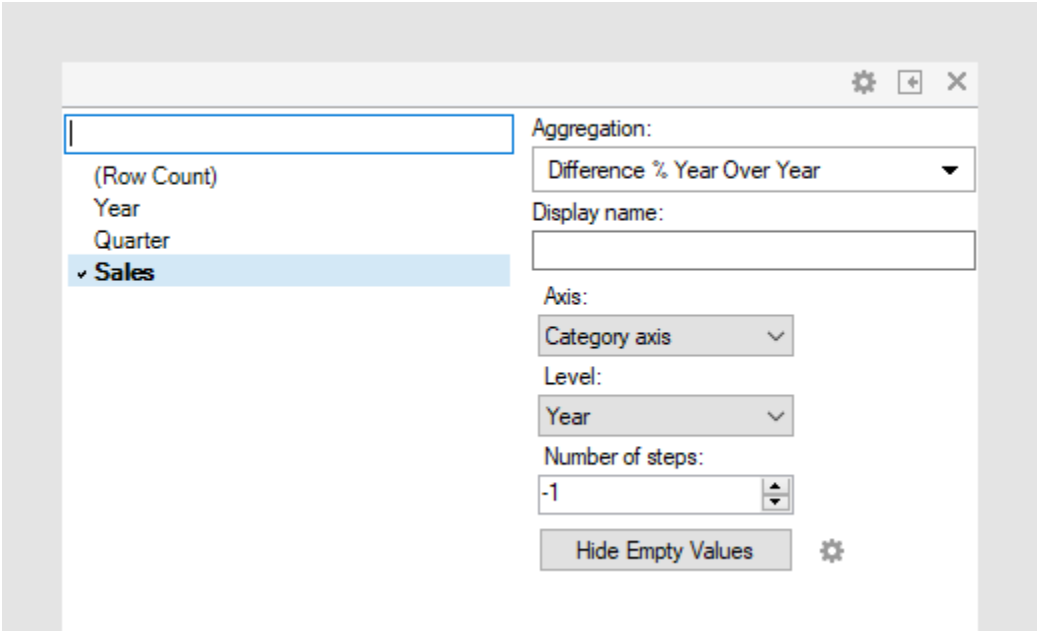
$-20\% = \$400 / \$500 - 1$

(see the rectangles below).




**The column selector in expanded mode**

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



Display name	Optionally, specify a different display name instead of the default "Difference % Year Over Year ([Column Name])".
Axis	<p>Specify the axis over which to calculate the nodes.</p> <p>Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.</p>

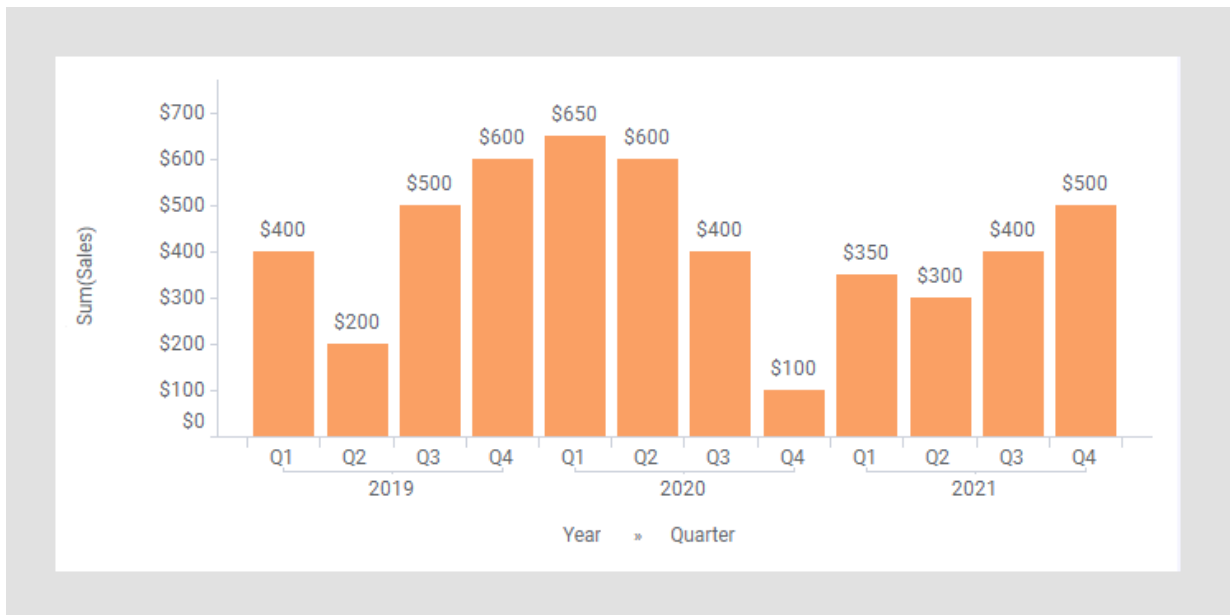
Level	<p>Define the hierarchy level to compare values at. You can either specify a named level such as "Year", "Quarter" or "Month", or you can set a number, which moves you up or down in the currently set hierarchy on the selected axis.</p> <p>If the level is 0 you will always compare the current leaf level, even if you change it in the visualization, for example, using a hierarchy slider.</p> <p>If the level is set to something other than "Year", then you are in fact doing a regular <a href="#">Difference %</a> calculation, and the aggregation selector will change correspondingly.</p>
Number of steps	<p>Define the number of steps between the nodes to compare. If the number of steps is -1, then the difference will be calculated between the current node and the one before it. If the number of steps is 1, then the difference will be calculated for the current node and the one following it.</p>
Hide Empty Values	<p>Creates a Show/Hide rule that hides all empty values. Click the properties symbol  to access the Show/Hide properties to edit or remove the rule.</p>

## % of Total

The % of Total expression shortcut calculates how many percent of the total each node contributes with.

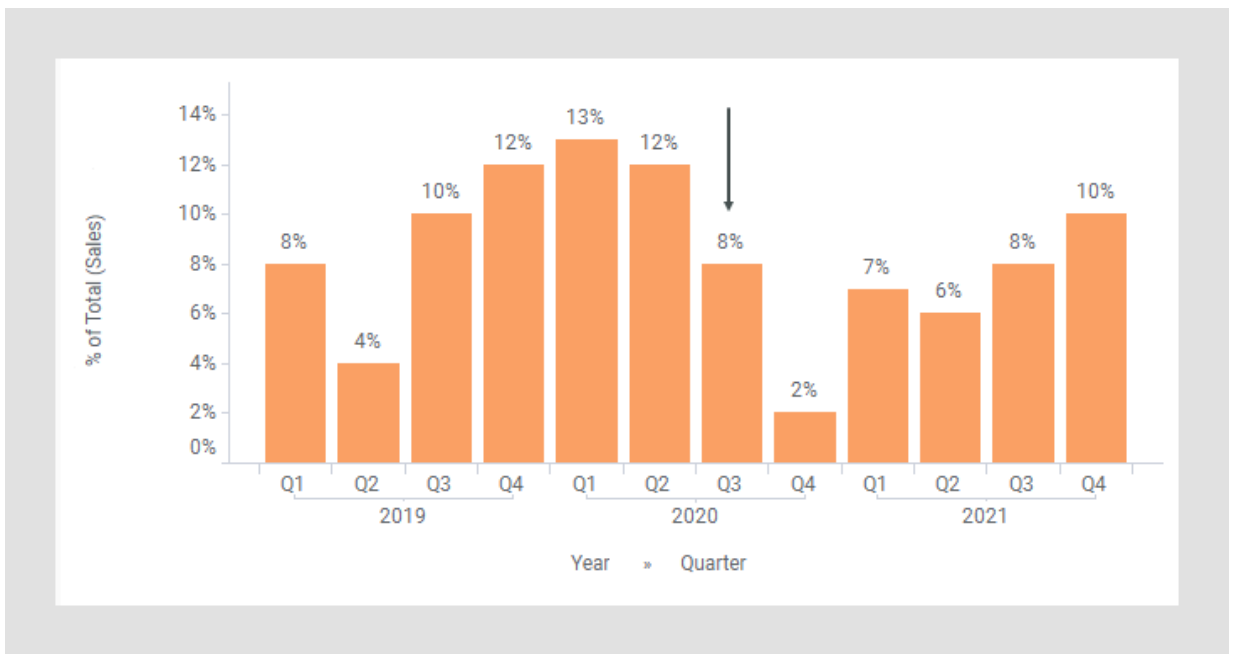
### Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression % of Total has been applied on the Value axis. (The arrow points to a bar used to explain what is calculated in the expression further down.)





The following expression is used on the Value axis.

```
Sum([Sales]) THEN [Value]/ Sum([Value]) OVER (All([Axis.X]))
```

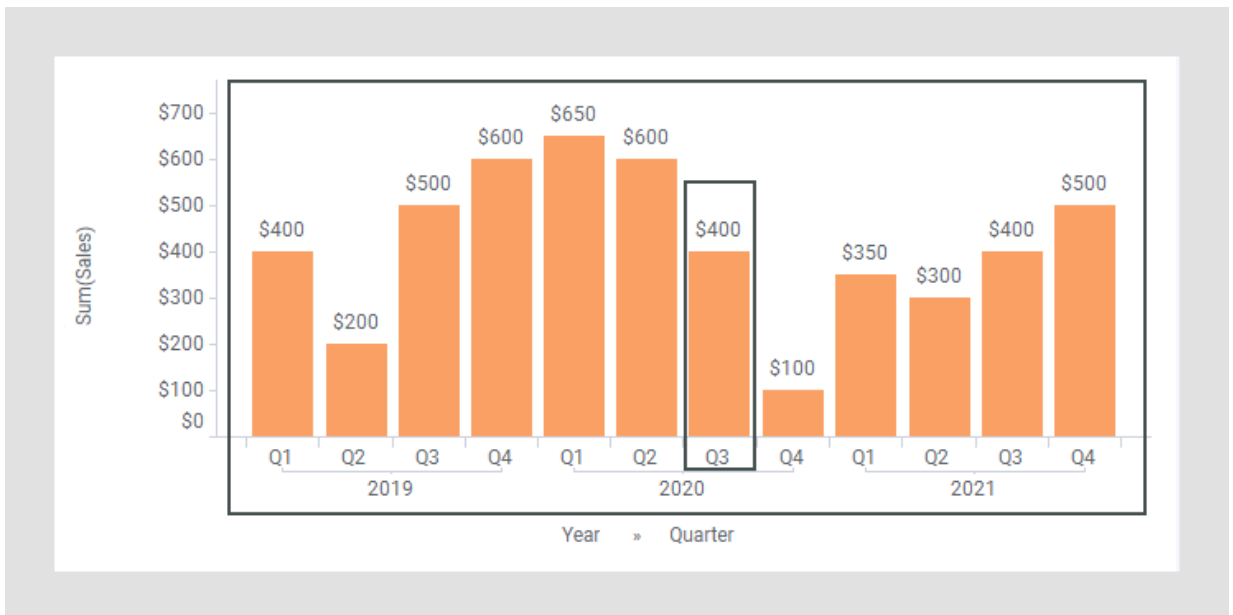
The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following THEN.

In this latter part of the expression, each node gets divided by the corresponding node the year before, because the number of steps is set to -1. If -1 is changed to -2, each node gets divided by the sum of all nodes.

For example, calculation of the % of Total for Q3, 2020 (see the arrow in the previous image):

$$8\% = \$400 / (\$400 + \$200 + \$500 + \$600 + \$650 + \$600 + \$400 + \$100 + \$350 + \$300 + \$400 + \$500)$$

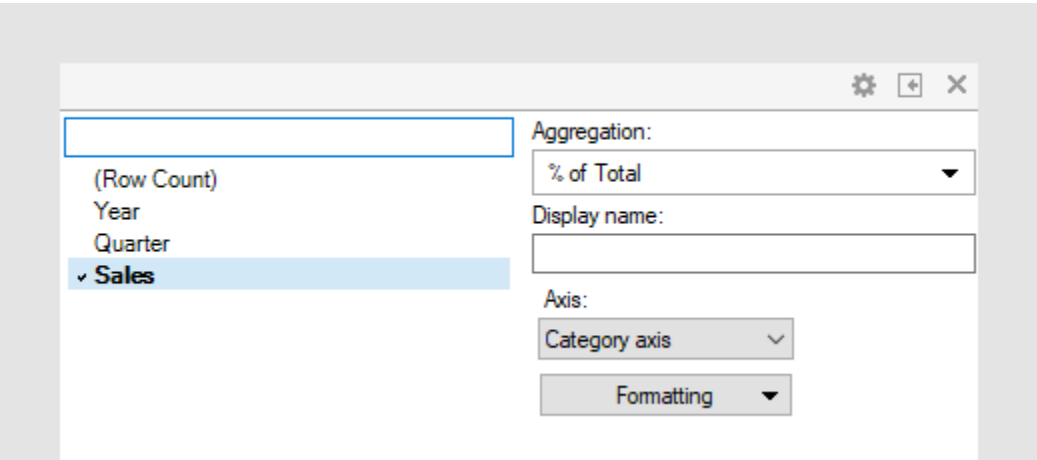
(see the rectangles below)



See also [All\(\)](#).

The column selector in expanded mode

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



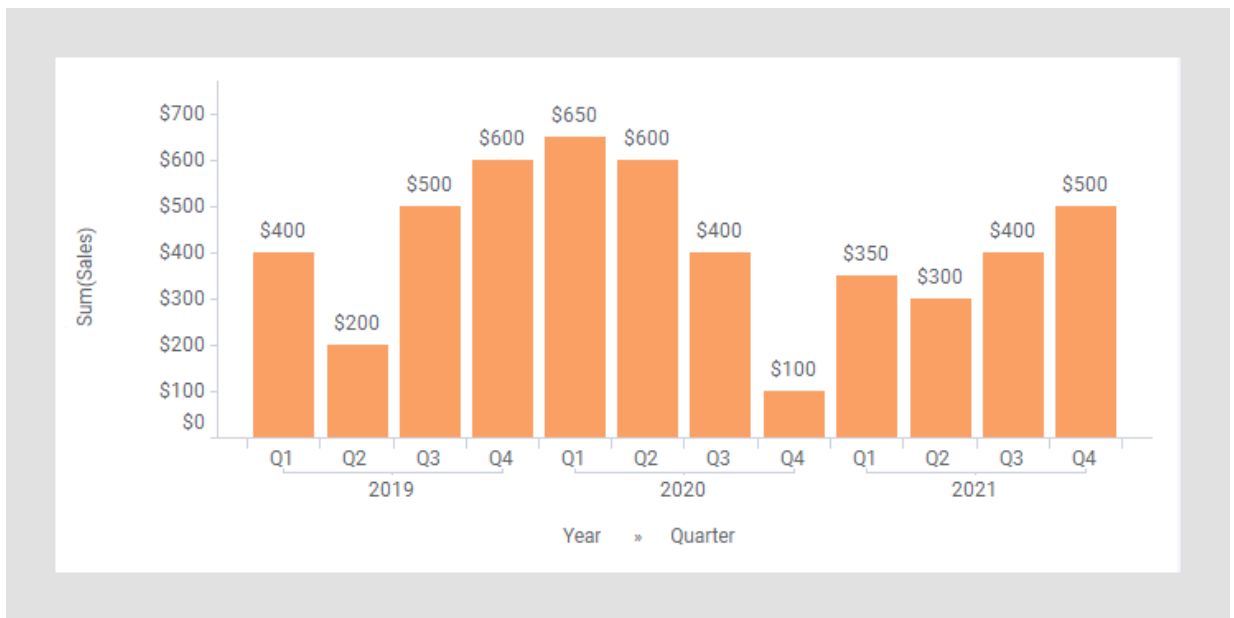
Display name	Optionally, specify a different display name instead of the default "% of Total ([Column Name])".
Axis	Specify the axis over which to calculate the nodes. Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.
Formatting	Change the formatting of the axis. For example, change to <b>Percentage</b> to show the values in percent rather than as decimal numbers.

Year to Date Total

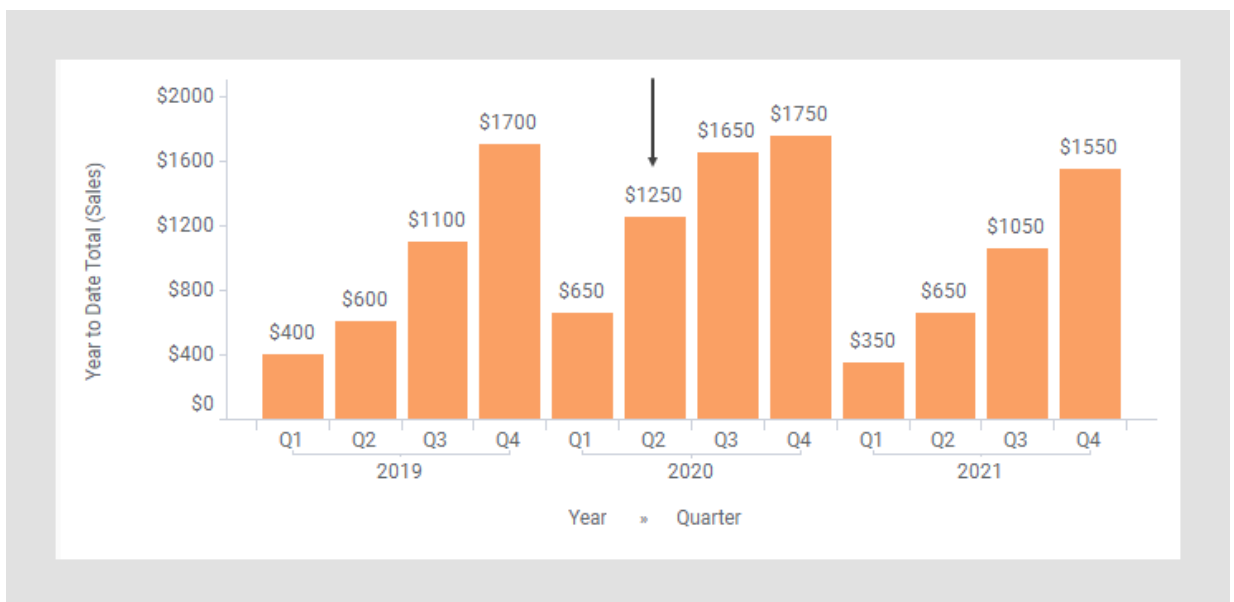
The Year to Date Total expression shortcut calculates the cumulative sum within each year. This shortcut is only applicable when you are having a time hierarchy including year on the X-axis.

Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression Year to Date Total has been applied on the Value axis. (The arrow points to a bar used to explain what is calculated in the expression further down.)



The following expression is used on the Value axis:

```
Sum([Sales]) THEN Sum([Value]) OVER (Intersect(AllPrevious([Axis.X]),
NavigatePeriod([Axis.X], "Year", 0.0)))
```

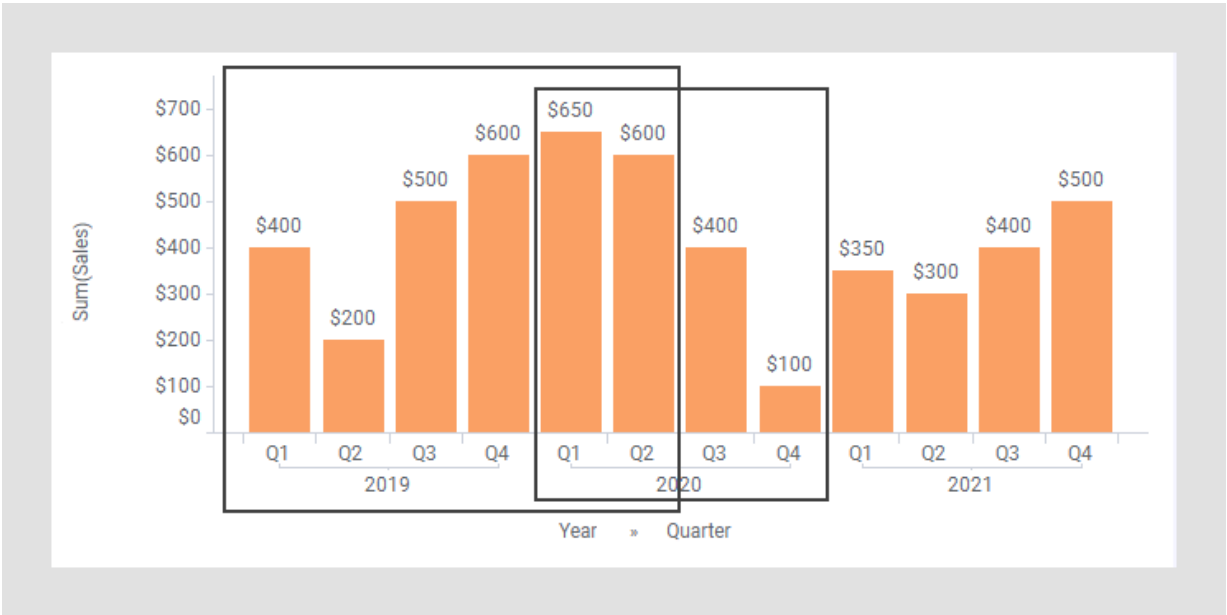
The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following THEN.

In this latter part, the nodes within an [intersection](#) of the results from two node navigation methods are summed; the [AllPrevious\(\)](#) method and the [NavigatePeriod\(\)](#) method. No matter which quarter you look at, the intersection will result in a period that begins with Q1 and ends with that quarter.

For example, calculation of the Year to Date Total for Q2, 2020 (see the arrow in the previous image):

$\$1250 = \$650 + \$600$

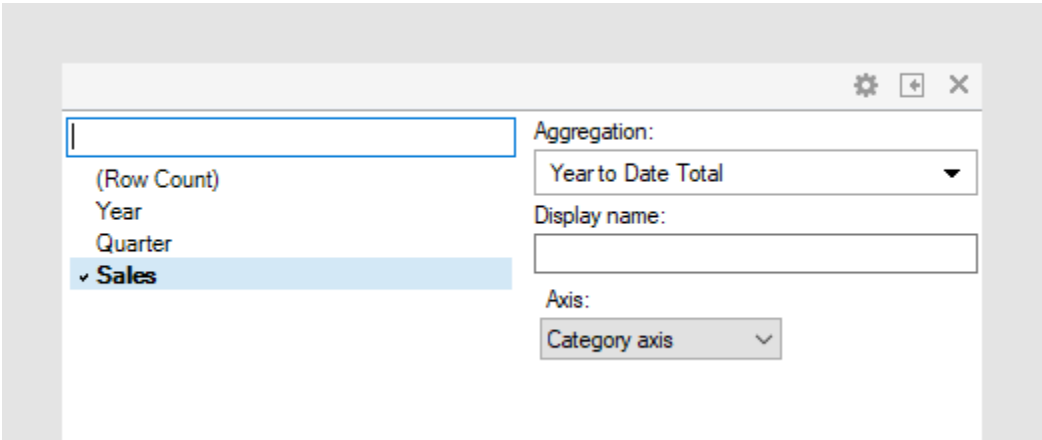
(see the left rectangle that represents `AllPrevious([Axis.X])` and the right rectangle that represents `NavigatePeriod([Axis.X], "Year", 0.0)`).



See also [Intersect\(\)](#).

### The column selector in expanded mode

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



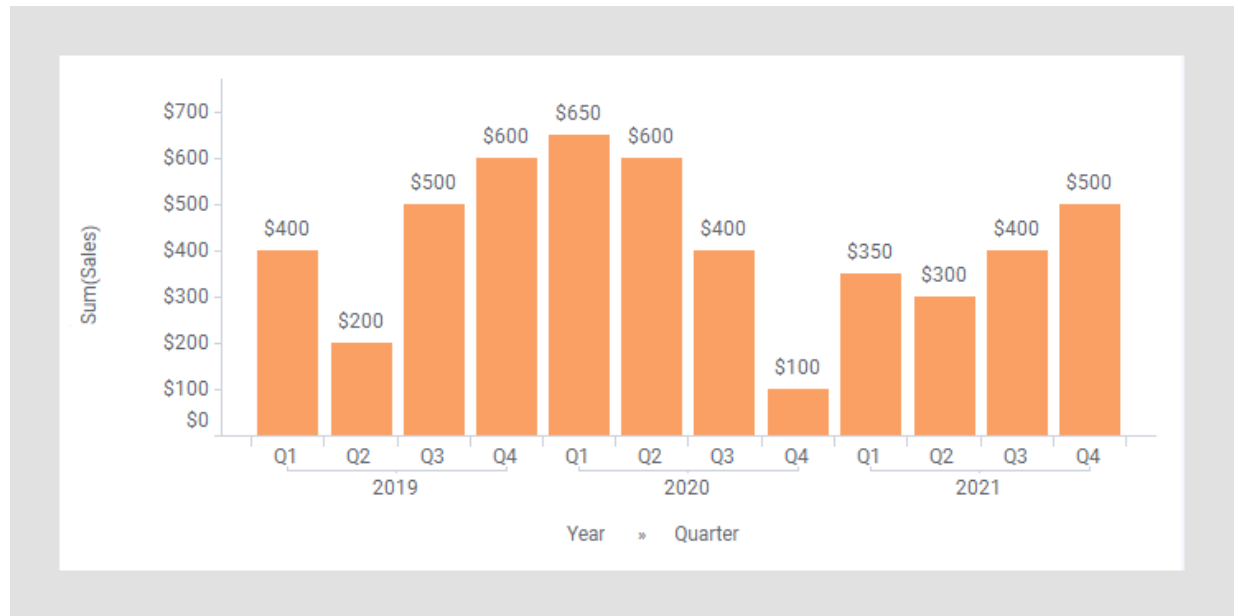
Display name	Optionally, specify a different display name instead of the default "Year to Date Total ([Column Name])".
Axis	<p>Specify the axis over which to calculate the nodes.</p> <p>Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.</p>

## Year to Date Growth

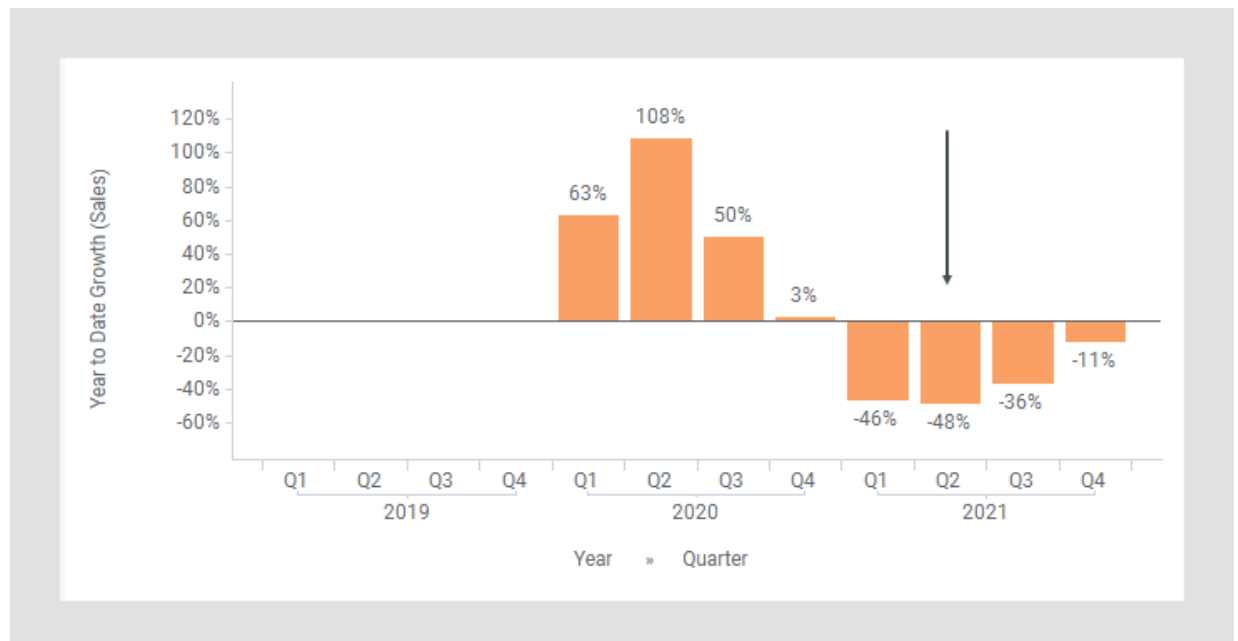
The Year to Date Growth expression shortcut calculates the relative difference between the cumulative sums for consecutive years. This shortcut is only applicable when you are having a time hierarchy including year on the X-axis.

### Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression Year to Date Growth has been applied on the Value axis. (The arrow points to a bar used to explain what is calculated in the expression further down.)



The following expression is used on the Value axis.

```
Sum([Sales]) THEN Sum([Value]) OVER (Intersect(AllPrevious([Axis.X]),
NavigatePeriod([Axis.X], "Year", 0.0))) THEN ([Value]/First([Value]) OVER
(NavigatePeriod([Axis.X], "Year", -1))) - 1
```

The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following THEN.

In the part between the THEN keywords, the nodes within the [intersection](#) of two node navigation methods are summed; the [AllPrevious\(\)](#) method and the [NavigatePeriod\(\)](#) method. The intersections will result in periods that begin with Q1 each year and end with the quarter to calculate the value for, that is, it is a [Year to Date Total](#) calculation.

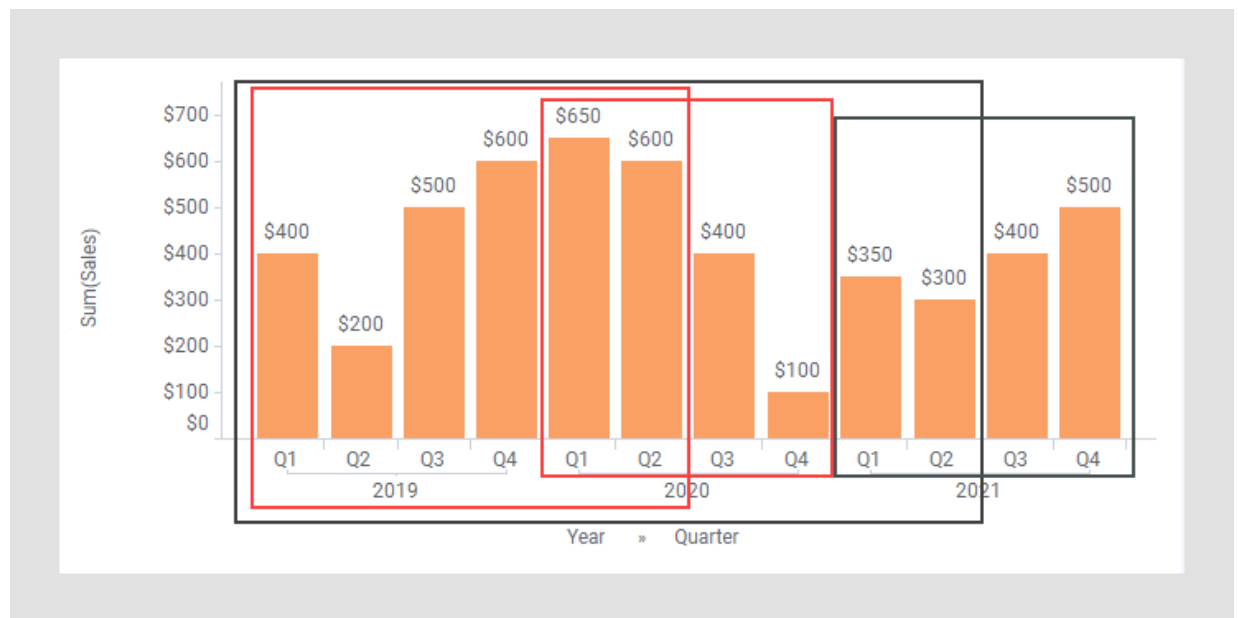
In the part after the second THEN, each node gets divided by the corresponding node one year ago (because the number of steps is set to -1, see [NavigatePeriod\(\)](#) for details). This result is then subtracted by 1 to get the percentage.

The aggregation First() is just used because an aggregation is required before the OVER keyword. It will take you to the first value of the node to subtract, but since each node only has one value, this corresponds to picking the value of the node.

For example, calculation of the Year to Date Total for Q2, 2021 (see the arrow in the previous image):

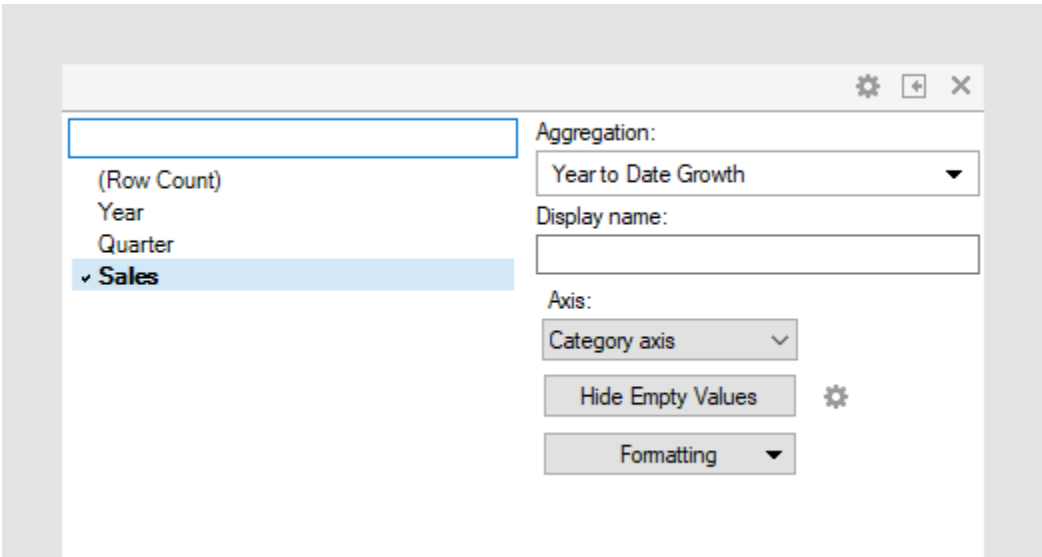
$$-48\% = (\$350 + \$300) / (\$650 + \$600) - 1$$


(See the bars within the intersection of the black rectangles. Adding them gives the year to date sum for Q2, 2021. In the same way, adding the bars within the intersection of the red rectangles gives the year to date sum for Q2, 2020. The percentage difference between those sums is then calculated.)



### The column selector in expanded mode

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



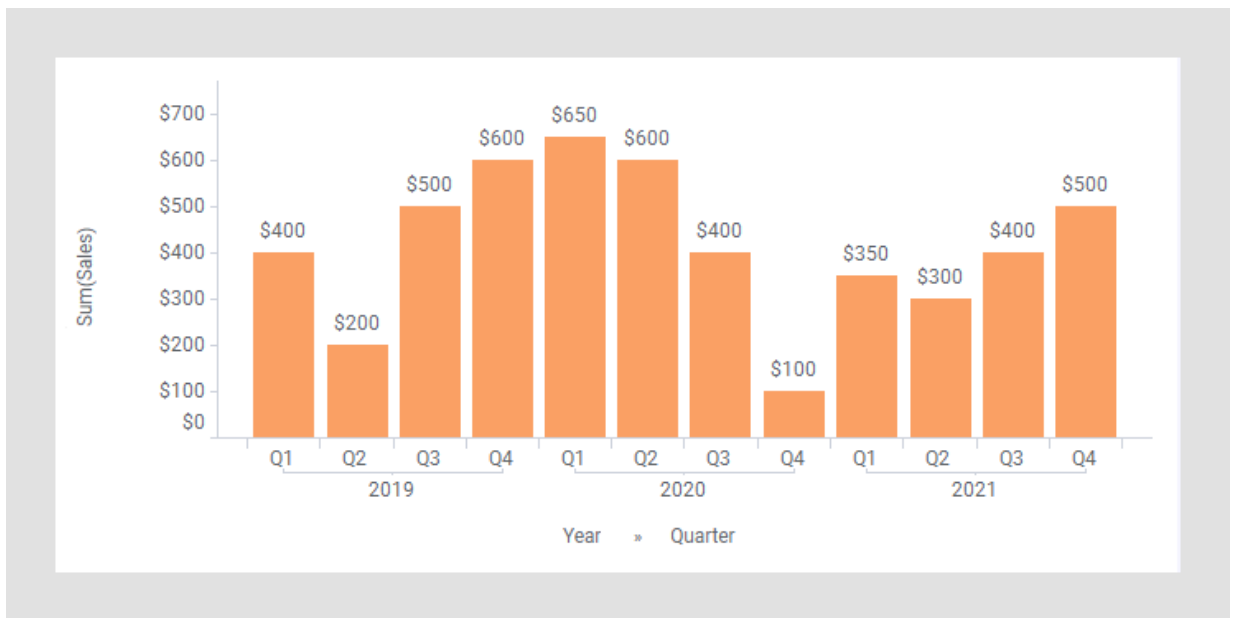
Display name	Optionally, specify a different display name instead of the default "Year to Date Growth ([Column Name])".
Axis	<p>Specify the axis over which to calculate the nodes.</p> <p>Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.</p>
Hide Empty Values	Creates a Show/Hide rule that hides all empty values. Click the properties symbol  to access the Show/Hide properties to edit or remove the rule.
Formatting	Change the formatting of the axis. For example, change to <b>Number</b> to show the values in numbers rather than as percentage.

**Change Relative to Start**

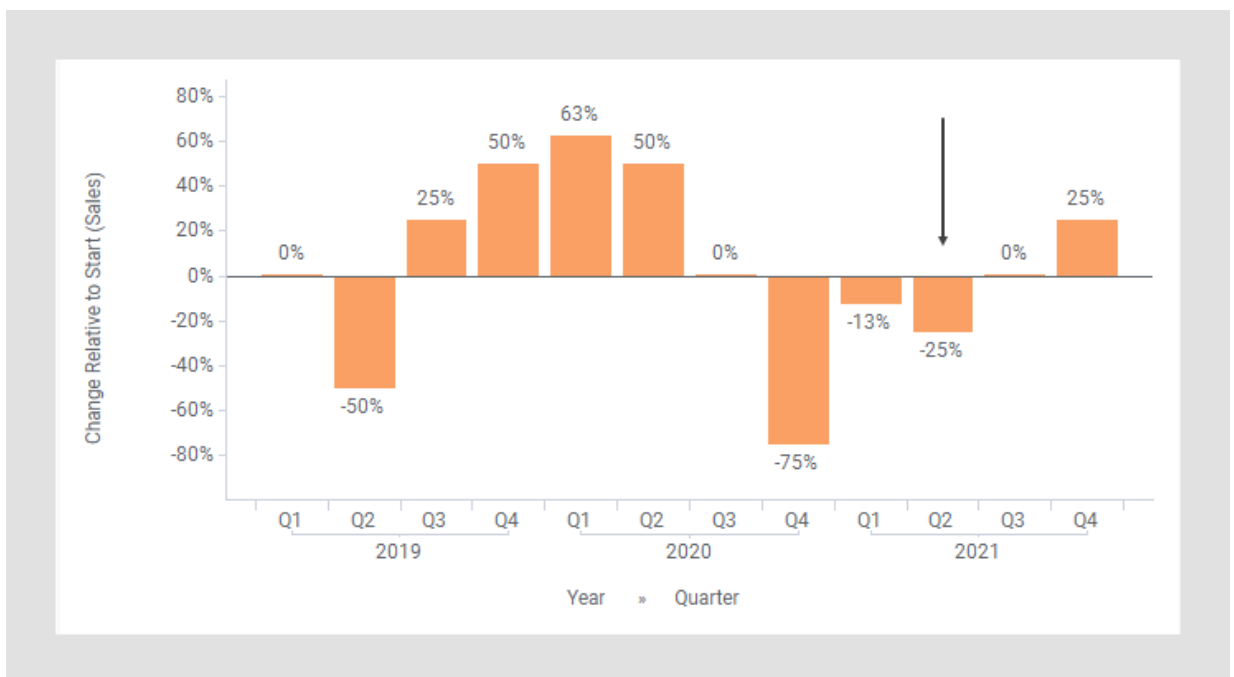
The Change Relative to Start expression shortcut calculates the percentage difference between a node and the first node.

**Example**

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression Change Relative to Start has been applied on the Value axis. (The arrow points to a bar used to explain what is calculated in the expression further down.)



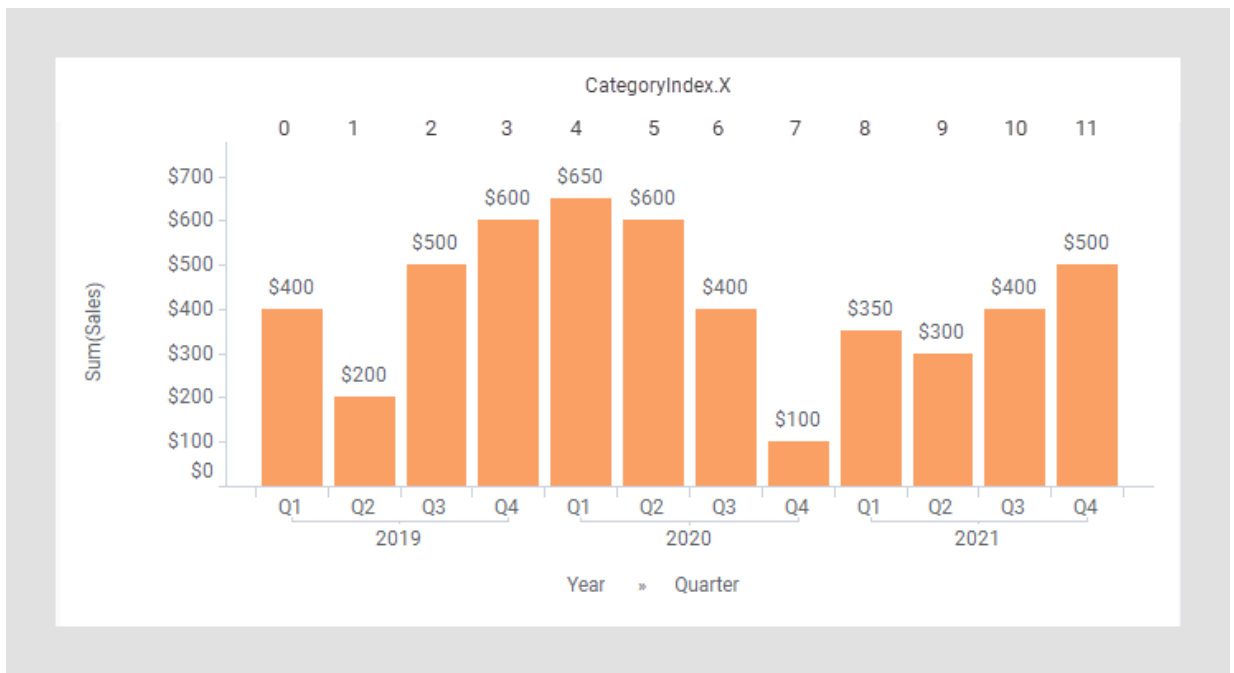
The following expression is used on the Value axis.

```
Sum([Sales]) THEN ([Value] / Sum(If([CategoryIndex.X]=0,[Value],0)) OVER
(All([Axis.X])))-1
```

The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following THEN.

In the part following THEN, each node gets divided by the first node. This result is then subtracted by 1 to get the percentage. To identify which node is the first one, and reference it, [CategoryIndex.X] is used. As illustrated in the image below, each node is allocated an index, starting with 0, so [CategoryIndex.X]=0 for the first node.



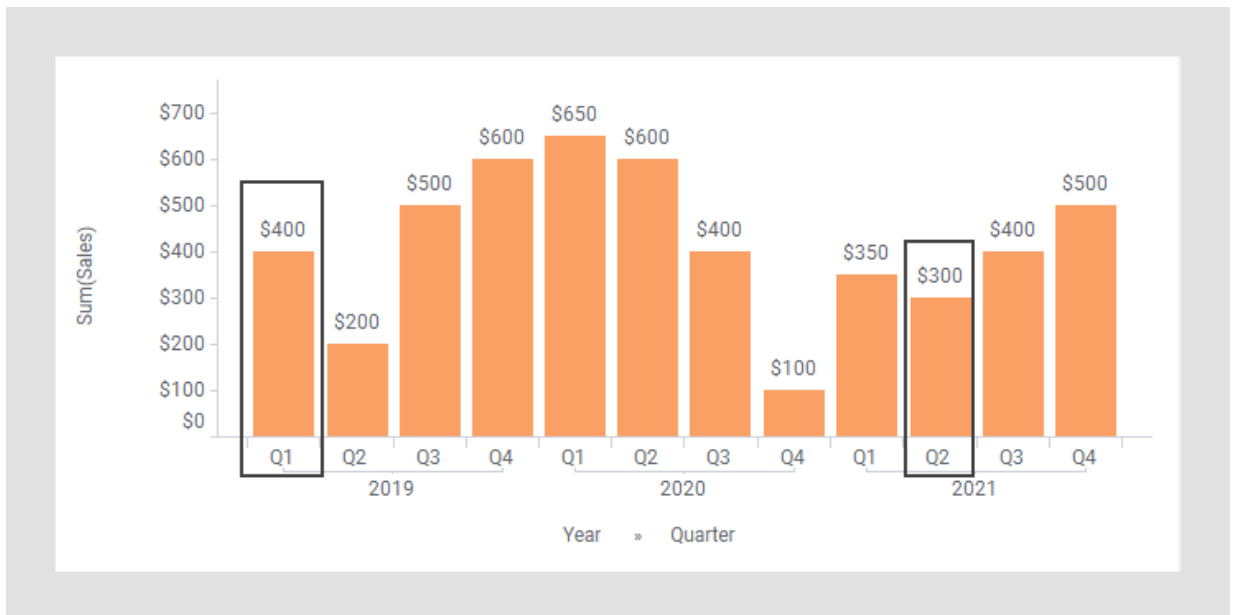


The `(If([CategoryIndex.X]=0,[Value],0)) OVER (All([Axis.X]))` expression finds and returns solely the the value of the node with `[CategoryIndex.X]=0`, in this case the \$400 value.

For example, calculation of Change Relative to Start for Q2, 2021 (see the arrow in the previous image):

$$-25\% = \$300 / \$400 - 1$$

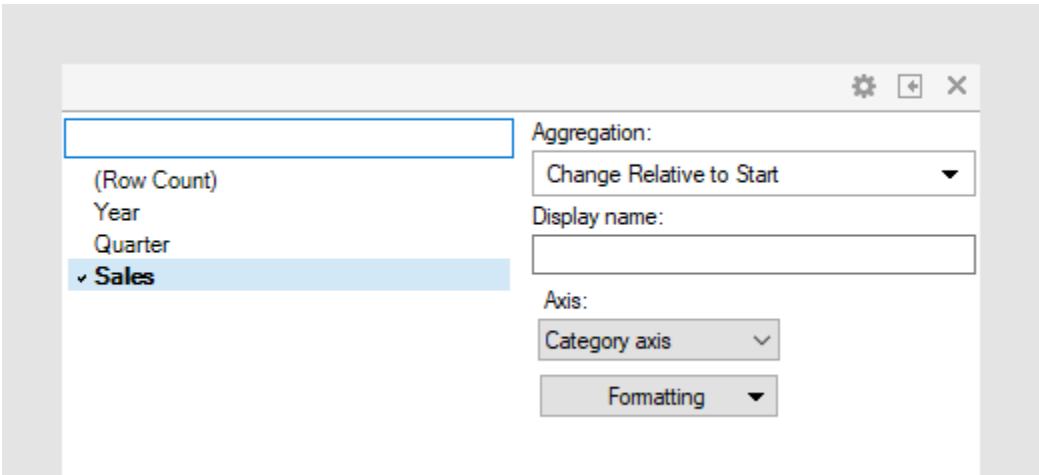
(See the rectangles below.)



See also [Change Relative to Fixed Point](#).

### The column selector in expanded mode

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



Display name	Optionally, specify a different display name instead of the default "Change Relative to Start ([Column Name])".
Axis	Specify the axis over which to calculate the nodes. Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.
Level	Change the formatting of the axis. For example, change to <b>Number</b> to show the values in numbers rather than as percentage.

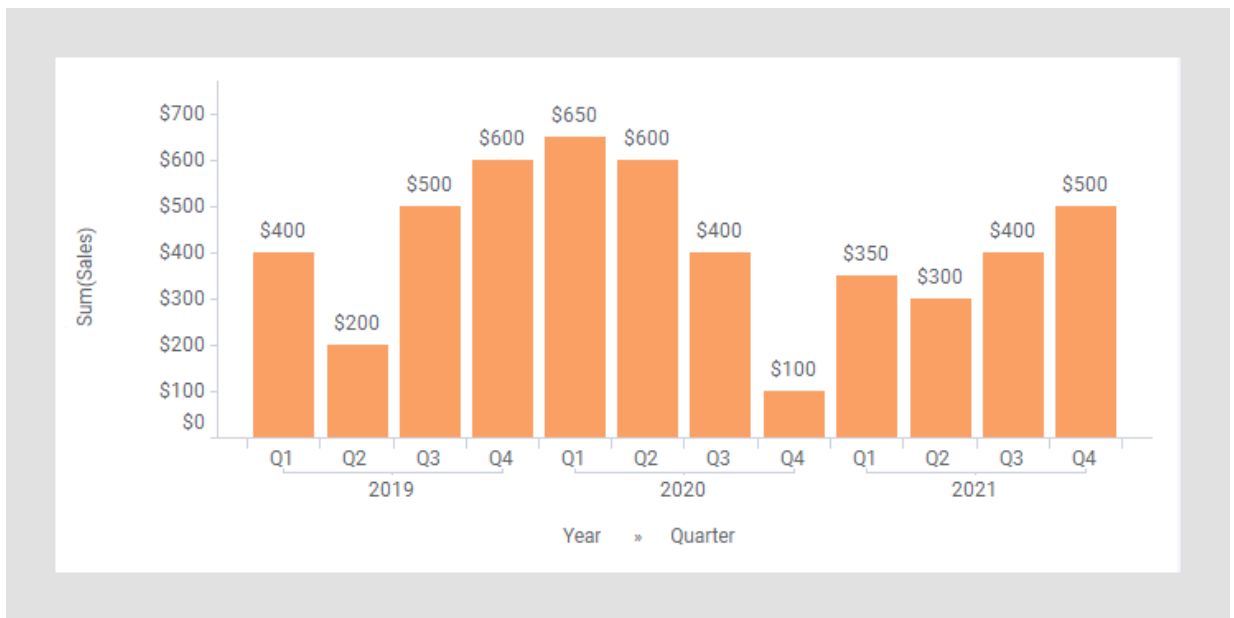
Change Relative to Fixed Point

The Change Relative to Fixed Point expression shortcut calculated the percentage difference between each of the nodes and a node of your choice.

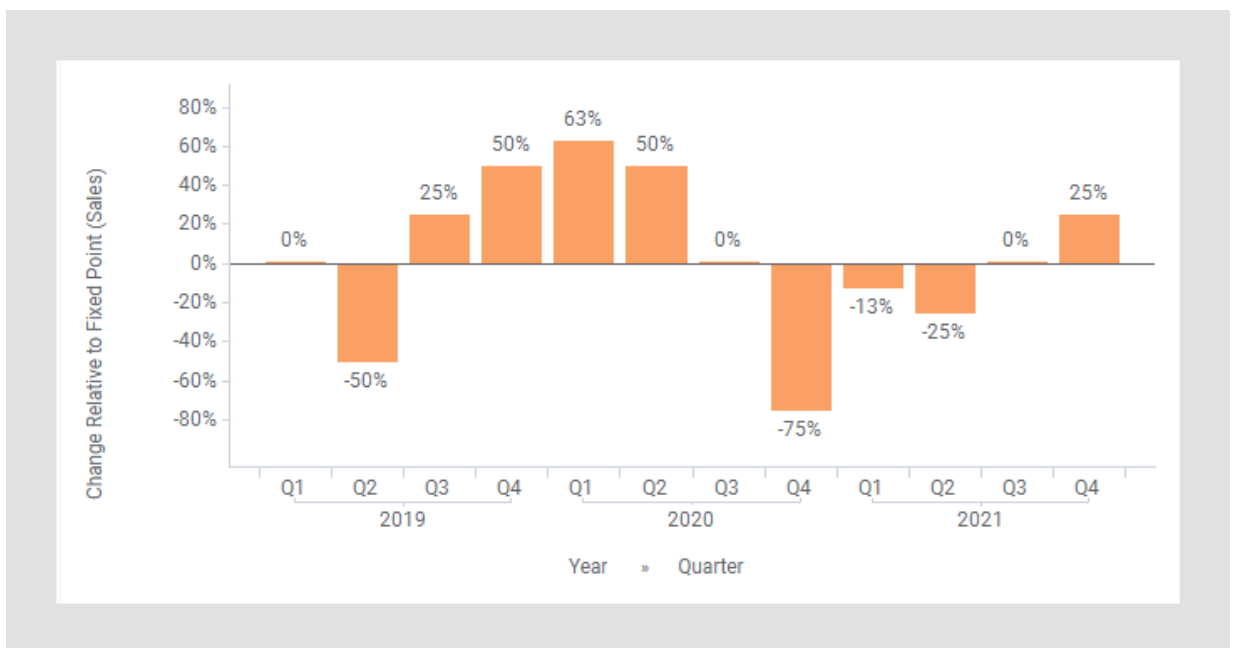
 The [Change Relative to Start](#) expression shortcut is a special case of this shortcut. When your choice is the first node, the result of the Change Relative to Fixed Point and Change Relative to Start will be identical.

Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression Change Relative to Fixed Point has been applied on the Value axis.

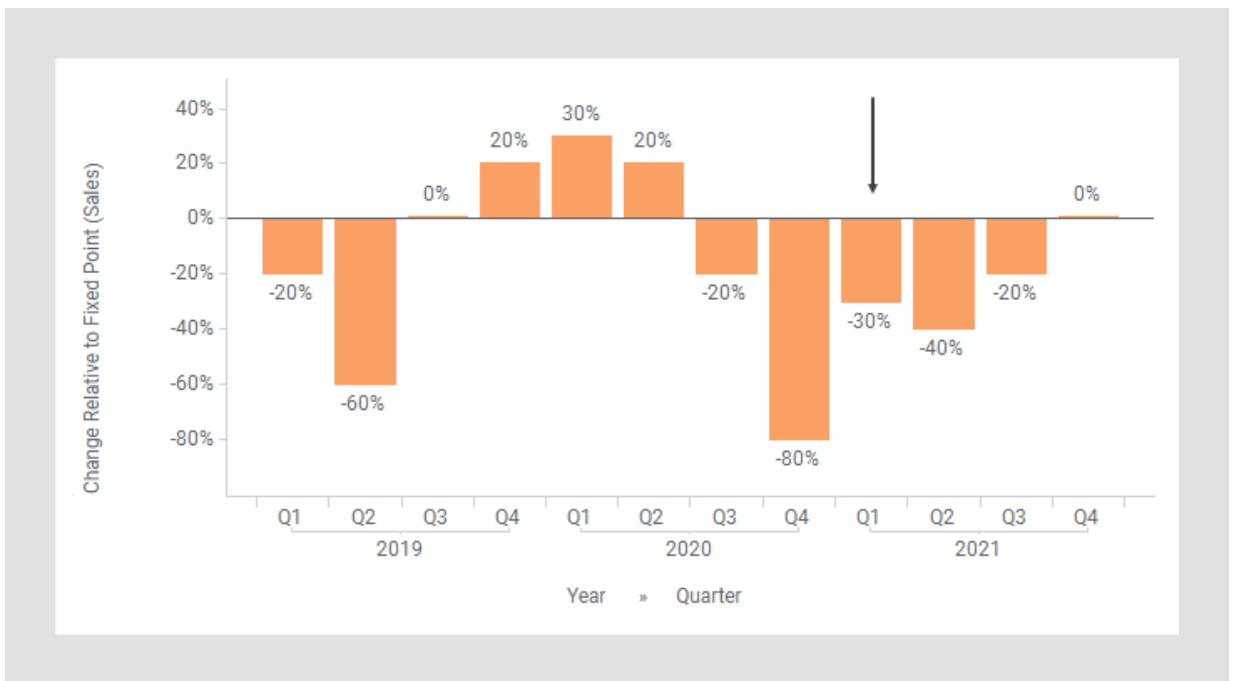


The following expression is in this case used on the Value axis. This default expression makes a comparison to the first node in the bar chart (Q1 in 2019).

```
Sum([Sales]) THEN ([Value] / Sum(If([X.Year]=2019) AND ([X.Quarter]="Q1",
[Value],0)) OVER (All([Axis.X])))-1
```

The settings can be changed to refer to any of the other nodes as fixed point. In the next bar chart, the expression is changed to make comparisons relative to Q3, 2019 instead. (The arrow points to a bar used to explain what is calculated in the expression further down.)

```
Sum([Sales]) THEN ([Value] / Sum(If([X.Year]=2019) AND ([X.Quarter]="Q3",
[Value],0)) OVER (All([Axis.X])))-1
```



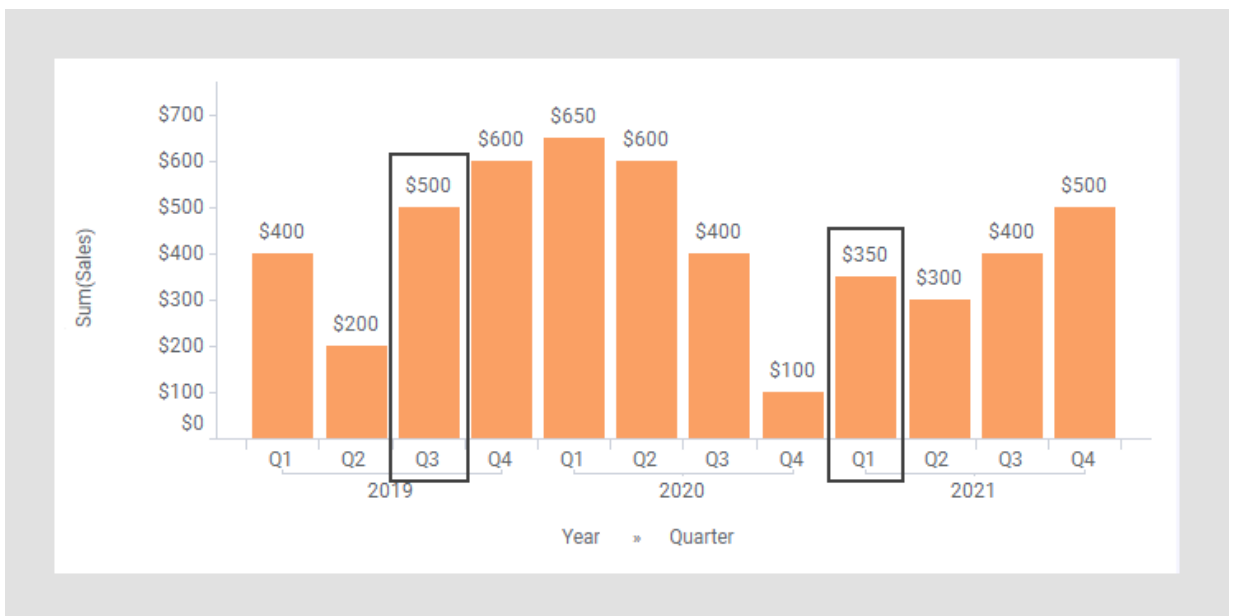
The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following THEN.

In the part following THEN, each node gets divided by the node specified to be the fixed one. This result is then subtracted by 1 to get the percentage.

For example, calculation of Change Relative to Fixed Point for Q1, 2021 (see the arrow in the previous image):

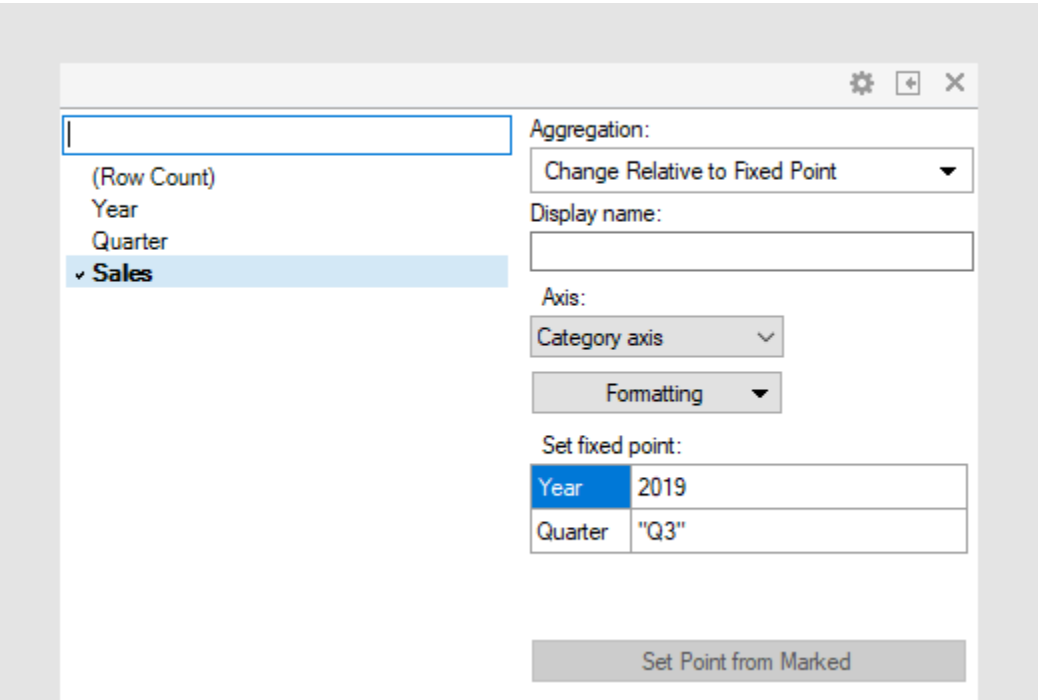
$$-30\% = \$350 / \$500 - 1$$

(See the rectangles below.)



The column selector in expanded mode

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



Display name	Optionally, specify a different display name instead of the default "Change Relative to Fixed Point ([Column Name])".
Axis	Specify the axis over which to calculate the nodes.  Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.
Formatting	Change the formatting of the axis. For example, change to <b>Number</b> to show the values in numbers rather than as percentage.
Set fixed point	Specify the fixed point against which the other nodes should be compared by typing a Year and a Quarter, or whatever you have on the hierarchy axis.
Set Point from Marked	Retrieves the point to compare against by marking a node in the visualization. (If multiple nodes have been marked, then the first node will be used.)

Compound Annual Growth Rate

The Compound Annual Growth Rate, CAGR, expression shortcut calculates the smoothed annual growth rate over a given time period.

To compare two investments, you can calculate the growth rates during a time period, for example, for measures such as revenue, production level, or number of registered users.

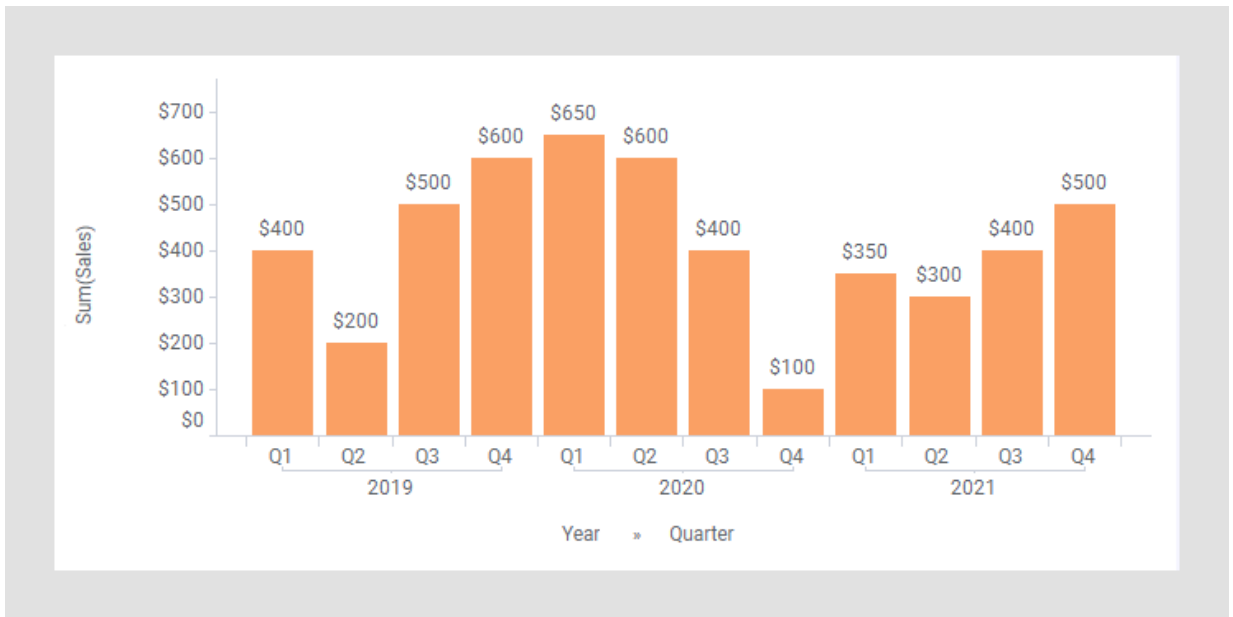
The formula used is

$$CAGR = \left(\frac{V_n}{V_0}\right)^{\frac{1}{n}} - 1$$

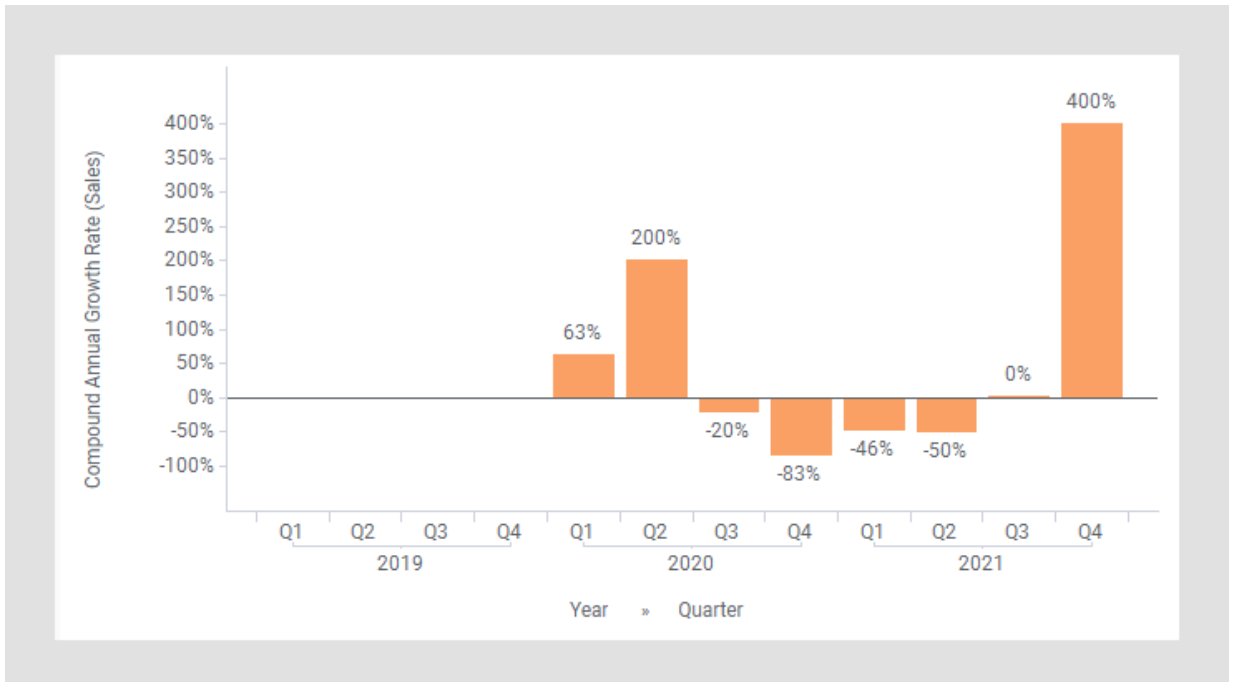
where  $V_n$  means the ending value,  $V_0$  the starting value, and  $n$  the number of years in the time period.

### Example

As starting point, the bar chart below is used. It shows the quarterly sums of sales for three years.



In the next bar chart, the shortcut expression Compound Annual Growth Rate has been applied on the Value axis.



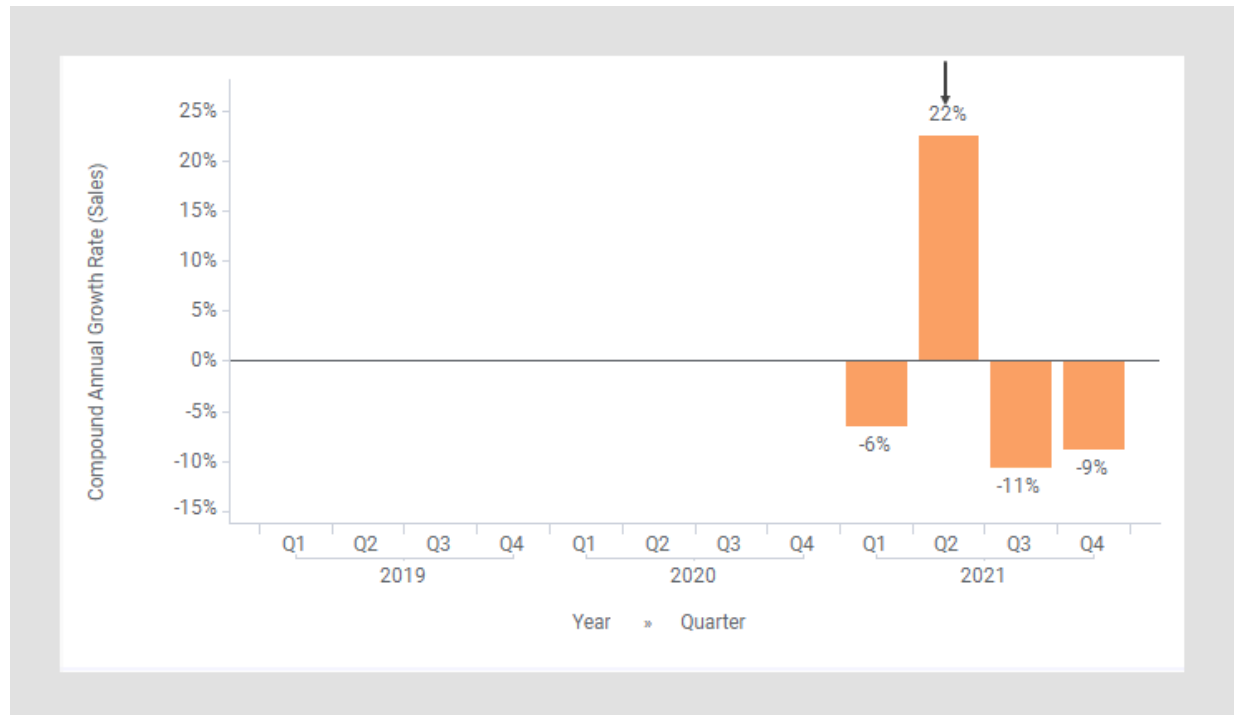
The following expression is used on the Value axis.

```
Sum([Sales]) THEN (Real([Value]/Sum([Value])) OVER
(NavigatePeriod([Axis.X], "Year", -1))) ^ (1/([X.Year] - First([X.Year]) OVER
(NavigatePeriod([Axis.X], "Year", -1)))) - 1
```

This default expression calculates the growth rate from one year to the next year, and not the rate over a number of years. To instead calculate the annual growth over, for example, two years, use the expression

```
Sum([Sales]) THEN (Real([Value]/Sum([Value])) OVER
(NavigatePeriod([Axis.X], "Year", -2))) ^ (1/([X.Year] - First([X.Year]) OVER
(NavigatePeriod([Axis.X], "Year", -2)))) - 1
```

This expression shows the growth rate, if the growth rate would be just as much each of the two years. The bar chart would change to the appearance below. (The arrow points to a bar used to explain what is calculated in the expression further down.)



The first part of the expression specifies that the column [Sales] should be aggregated as a sum. The result of this part goes into the [Value] column to be used in the expression following THEN.

After THEN comes first a conversion to the data type Real to make sure that ^ (the power function) will work for all column types, for example, Currency.

Each node is then divided by the corresponding node two years earlier, and this ratio is in turn raised to  $\frac{1}{2}$ . ( $\frac{1}{2}$  is because the number of years is two. When the number of years is three, the ratio is raised to  $\frac{1}{3}$ .) Then the result is subtracted by 1 to get the percentage.

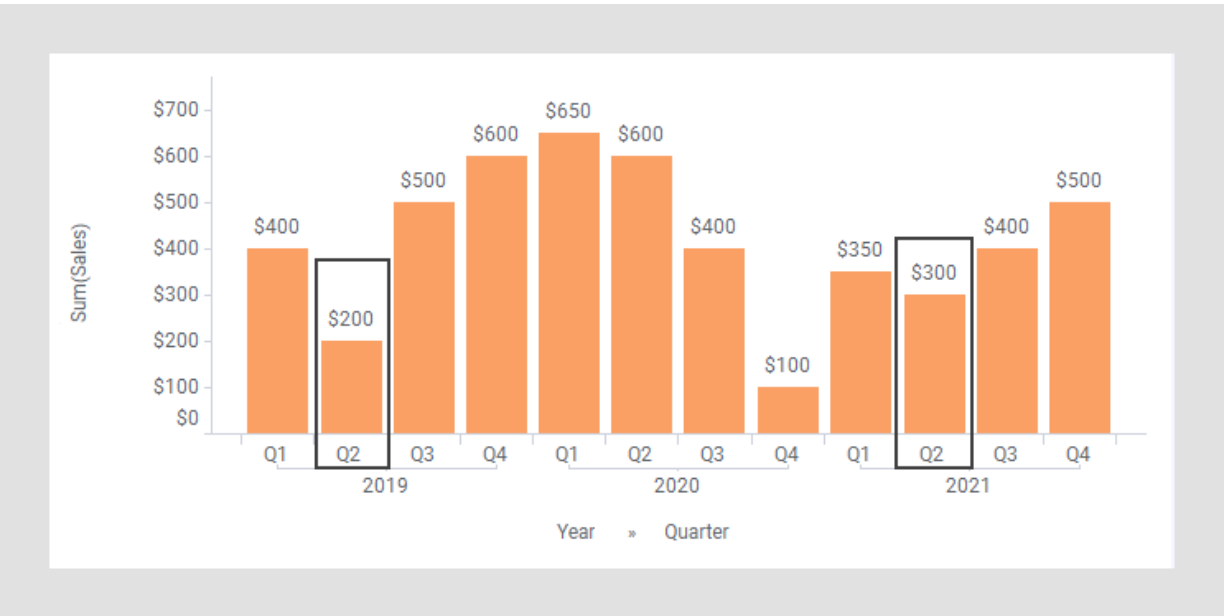
The aggregation First() is just used because an aggregation is required before the OVER keyword. It will take you to the first value of the node to subtract, but since each node only has one value, this corresponds to picking the value of the node.

For example, the calculation of the Compound Annual Growth Rate for Q2, 2021:

$$22\% \approx (300\$ / 200\$)^{1/2} - 1 = \sqrt{\frac{300}{200}} - 1$$

This means that the smooth annual growth rate is 22%, that is, after two years, the amount is  $200 * 1.22 * 1.22 \approx 300$ .

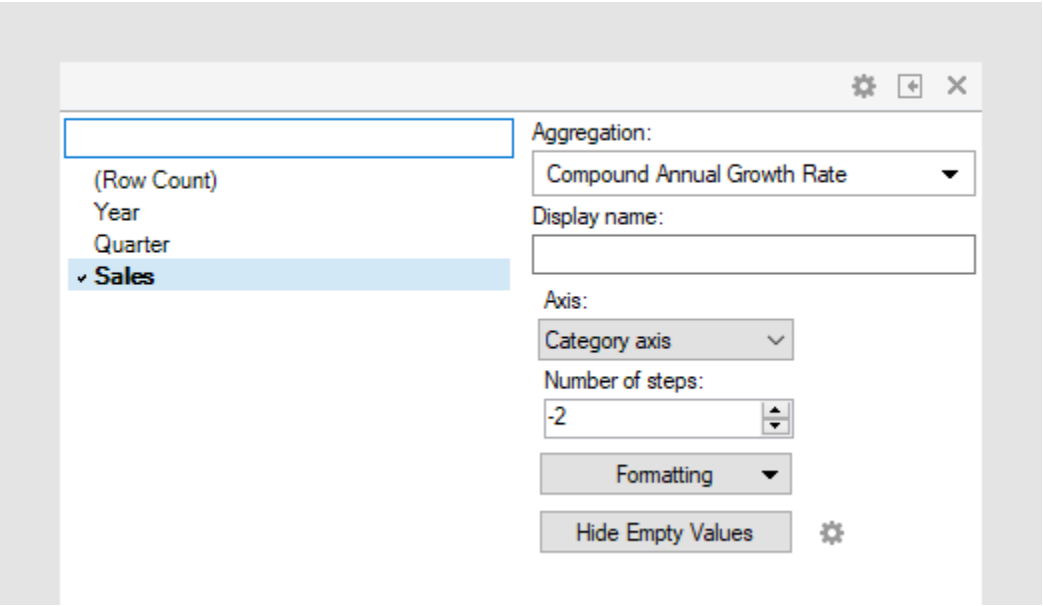
(See the rectangles below.)



See also [NavigatePeriod\(\)](#).


**The column selector in expanded mode**

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



Display name	Optionally, specify a different display name instead of the default "Compound Annual Growth Rate ([Column Name])".
Axis	<p>Specify the axis over which to calculate the nodes.</p> <p>Only categorical axes that perform some kind of grouping can show up in the drop-down list. This means that if you only have a grouping on the X-axis, then this is the only axis available, whereas if you also have colored by a categorical column then the Color axis will be available as well, and so on.</p>



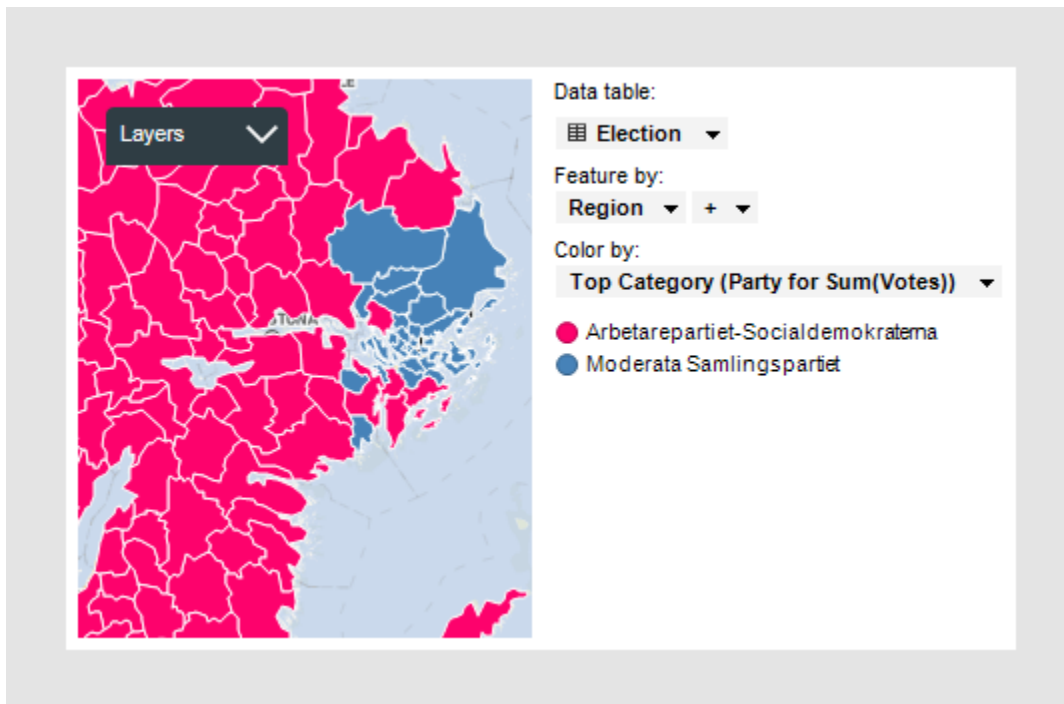
Number of steps	Define the number of steps between the nodes to compare. If the number of steps is -1, then the difference will be calculated between the current node and the one before it. If the number of steps is 1, then the difference will be calculated for the current node and the one following it.
Formatting	Change the formatting of the axis. For example, change to <b>Number</b> to show the values in numbers rather than as percentage.
Hide Empty Values	Creates a Show/Hide rule that hides all empty values. Click the properties symbol  to access the Show/Hide properties to edit or remove the rule.

## Top Category

The Top Category expression shortcut is used to quickly visualize a winning category in a categorization visualization. For example, it can be used for coloring a map chart with winning political parties or products.

### Example

Below is a feature layer map showing the different regions in Sweden. It is colored by the top category political party, based on the number of votes that each party got in that region.



The expression used on the **Color by** axis:

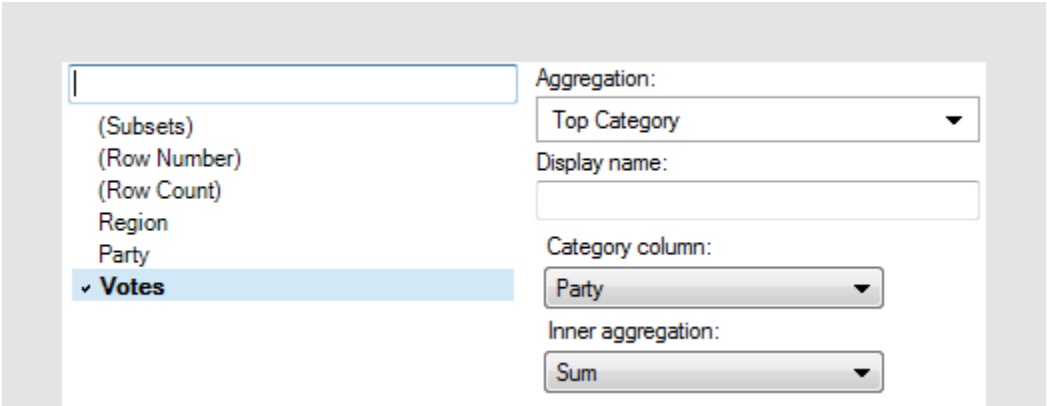
```
UniqueConcatenate(If(Max(Sum([Votes]) OVER (Intersect([Party]))) OVER
(Intersect())=Sum([Votes]) OVER (Intersect([Party])),[Party],null))
```

The inner aggregation is what is calculated first: in this example it is the sum of votes per party. The number of votes is split by region as defined by the categorical axes in the visualization only (handled by the empty Intersect() statement). Then, the party with the max number of votes is found, and if the smeared number of votes value for a row equals the max number of votes, the name of that party is sent along to the next step, otherwise the result is null, since parties with a sum(Votes) lower than max are not interesting. All parties that have the max number of votes are concatenated to a list so that ties between several parties with the max number of votes are handled. In order to avoid the same party name being shown more than once, it is the unique concatenate function that is applied to the result.

See also [Intersect\(\)](#).

**The column selector in expanded mode**

In the installed client, the extended column selector offers controls that make it easy to change various parameters in the expression shortcut:



Display name	Optionally, specify a different display name instead of the default "Top Category ([ Category Column Name] for InnerAggregation([ Column Name]))".
Category column	Specify the categorical column to find the winner in.
Inner aggregation	Specify the aggregation method of the inner aggregation where you want to find the winner. The default aggregation is sum.

**Expression language details**

If you learn how the expression language works, the creation of valid expressions will be easier. This topic attempts to explain some basic functionality of the language.

If a column name contains characters other than letters, digits or \_ (underscore), then they must be enclosed with "[" and "]" characters (square brackets). The square brackets are also required when the column name is the same as a function name, or when the column name begins with a digit. For example:

```
[Column 1]
```

```
[Sum]
```

If you apply <> around an expression, it is treated as categorical. For example:

```
<[Category column 1]>
```

Multiple columns are normally separated by a comma. For example:

```
[Column1],[Column2]
```

You can use the AS keyword to specify a new name for an expression, and hide the underlying expression. For example:

```
[Column1]/([Column1]+[Column2])AS Quota
```

Note that when the axis expression contains commas, you cannot rename the entire axis, but you can define one name for each part of the expression:

```
[Column1] AS [My first column], [Column2] AS [My second column]
```

Constants can always be used as an argument, even if the function description says that the argument has to be a column.

```
39+12*3
```

```
-(1-2)
```

If you are working with visualizations based on more than one data table, expressions will refer to the main data table by default. To refer to a column in another data table, you must use the qualified name: [Data Table Name].[Column Name]. For example:

```
Sum([Data Table 2].[Column1])
```

See also [Operators](#) and [Functions](#) for complete descriptions of the available operators and functions.

### Comments in expressions

If you create more complex expressions that need an explanation, it is possible to add comments to the expressions. Use `//` to specify that the text after the symbols is a comment.

For example:

```
Avg([Sales]) OVER Intersect([Cat], AllPrevious([Year]))
//Shows the average sales per category for all years, up until the current year.
```

```
[Sales]/1000 //The sales figure has been divided by a thousand to show values in kkr instead
of SEK
```

Note that if you add comments to an expression on a visualization axis, it is expected that you continue using the custom expression dialog for all further editing. This is because the drag-and-drop operations affect and rearrange the expressions too much to be able to keep comments.

### Escaping special characters

Because certain characters have a special meaning in the Spotfire expression language, you must perform some actions if you are going to use those characters for other purposes, such as including them in column names.

An escape character is a special character used to inform the expression language that the following character in a character sequence should be seen as a standard character rather than as an item performing its special purpose.

If a column name contains any square brackets then they must be escaped. Escaping of brackets is performed by adding extra brackets before and after the bracket: "[ " and " ] ]". For example, if the column name is [Name], it would be written as [ [Name] ] ] in an expression. When using the **Insert Column** button in the Add calculated column dialog (installed client only) to add columns, you will automatically get the correct escaping of the column name.

### Case sensitivity

- Variables, functions and keywords are not case sensitive: SUM(C1) = Sum(C1) = sum(C1)
- Column name references are case sensitive.

- Method call names are not case sensitive. All methods which are defined in the add-in framework can be used. See [Operators](#) and [Functions](#) for information about the different methods supported.

## Expression results

An expression describes how a new column should be calculated. The newly created column will have the same number of rows as all the other columns in the data table. The default null handling behavior is that operations on null return null. This means that if a new column is calculated as `[Column A]*2` and there are empty values on some rows in Column A, then the new column will have empty values on those rows as well.

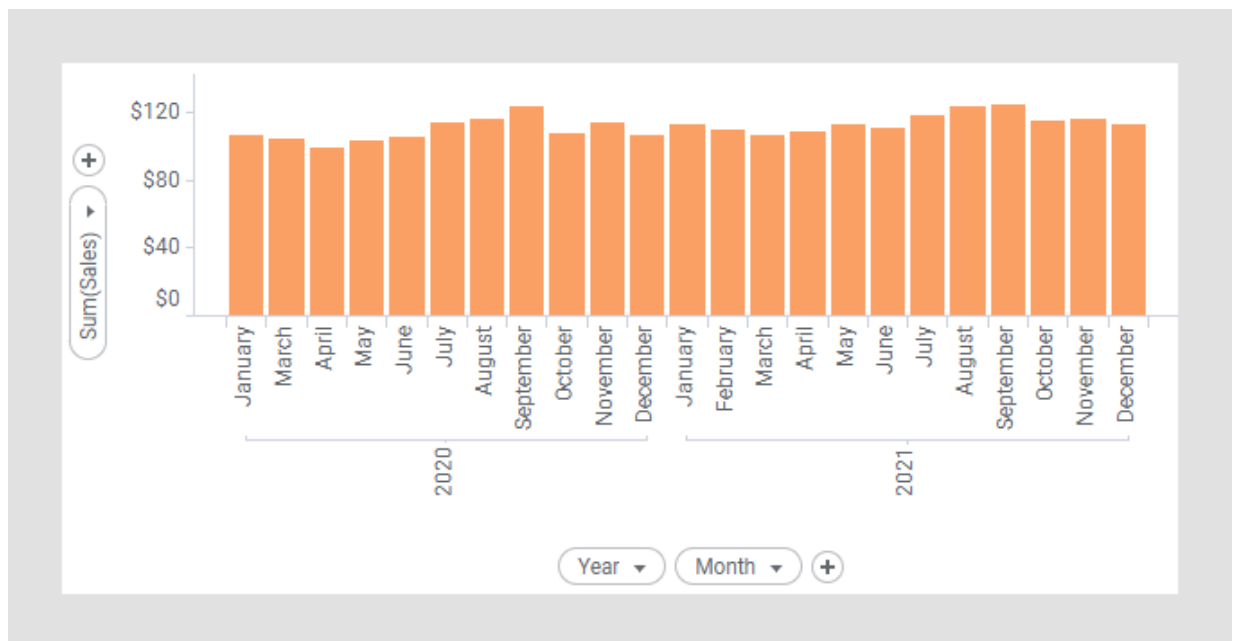
## Categorical expressions, NEST and CROSS

In custom expressions, categorical and hierarchical columns and expressions are written between angles, "<>". When more than one category is available within the expression, which combinations of categories to show must also be specified. This is done using the keywords NEST (which shows all actual combinations of values in the data) or CROSS (which will show all possible combinations of the values, including combinations that currently hold no data). All columns in the expression must be separated by "nest" or "cross" instead of a comma, and mixing the two combination options is not permitted.

```
<[Category column 1] nest [Category column 2]>
```

For example, if you have a data table containing some sales data for each month during two years, but the data for February is missing for 2020, the different options will give the following results:

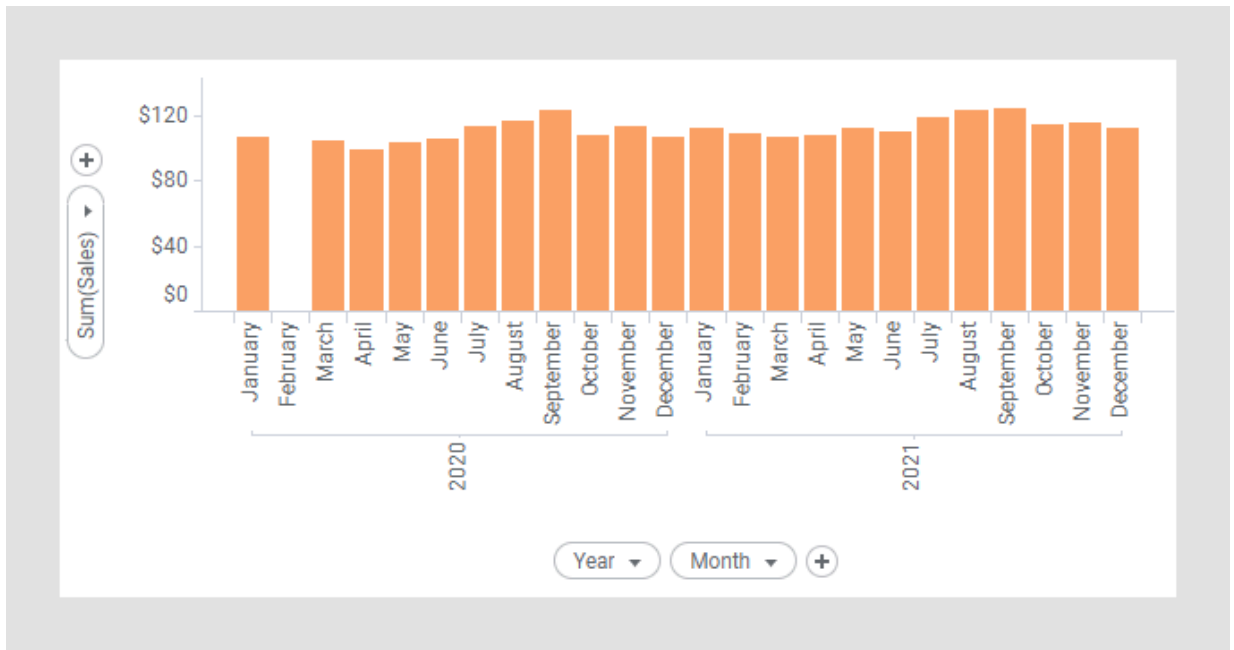
### Nest



Because there is no data available for February 2020, there will not be a bar (nor a placeholder for a bar) there. This visualization is set up using the Nest option on the category axis and it corresponds to the custom expression:

```
<[Year] NEST [Month]>
```

### Cross



When the CROSS option is selected, all possible combinations of the categories can be displayed. This means that there will be a placeholder for the February column for 2020, even though there is no data available for February. The corresponding custom expression would be:

```
<CROSS [Year] CROSS [Month]>
```

(The first CROSS is optional.)



Even though the CROSS option is selected, you might not see the empty categories in the visualization depending on the axis mode. In the installed client, you can right-click on the axis and select **Categories** to view or change the settings. If **Show filtered values** is selected, then categories without any values will be hidden in the visualization. You must use **Show filtered range** or **Show all values** to see the empty category in this example.



In the installed client, the setting for NEST/CROSS, as well as the settings for the axis mode and how to evaluate the axis expression, can also be specified from the **Visualization Properties > Category Axis > Advanced Settings** dialog.

## Validation of values

Empty values are generated whenever a column value from the data table is missing, when a calculation involves an invalid value, or by explicitly writing `null` in the expression. Results that are `null`, are displayed as (Empty) or simply left blank. Null values can be replaced using the [SN\(\) function](#).

When aggregating within a column, the invalid value will be ignored, whereas row-wise calculations between columns will result in invalid values each time one of the involved columns contains an invalid value.

## Operators

A list of the operators you can use in custom or calculated expressions.

Operator	Description
- Arg1	Negates the argument. The argument and the result are of type Real.

Operator	Description
$Arg1 + Arg2$	Adds the two arguments. See operator & for String concatenation.
$Arg1 - Arg2$	Subtracts $Arg2$ from $Arg1$ .
$Arg1 * Arg2$	Multiplies the two arguments. The arguments and the result are of type Real or Decimal.
$Arg1 / Arg2$	Divides $Arg1$ by $Arg2$ . The arguments and the result are of type Real or Decimal. Division by zero results in an invalid value.  Examples: $7/2 \rightarrow 3.5$ $0/0 \rightarrow (Empty)$ $-1/0 \rightarrow (Empty)$
$Arg1 \& Arg2$	Appends $Arg2$ to the end of $Arg1$ . The arguments can be of any type, but are converted to strings. The result is of type String. See also function Concatenate.  Examples: $"April " \& (20+1) \& "st" \rightarrow "April 21st"$ $null \& "Ape" \rightarrow (Empty)$
$Arg1 \% Arg2$	Returns the remainder of the division of $Arg1$ by $Arg2$ . The arguments and the result are of type Real or Decimal. Invalid values are propagated to the result column.  Example: $3.5 \% 2.5 \rightarrow 1.00$
$Arg1 \neq Arg2$	Returns true if $Arg1$ is not equal to $Arg2$ .  Examples: $If( 1 \neq 2, true, false )$ $Case\ when\ 2 \neq 2\ then\ true\ else\ false\ end$
$Arg1^{Arg2}$	Returns $Arg1$ raised to the $Arg2$ power.  Example: $2.5^3$ $[Value\ Column]^2$
$Arg1 < Arg2$	Returns true if $Arg1$ is less than $Arg2$ . The arguments can be of any type, but must both be of the same type. The result is of type Boolean. If any argument is invalid, the result is invalid. The function is defined for comparing normal numbers to each other. Other combinations result in invalid values.  Examples: $If( 1 < 2, "true", "false" ) \rightarrow true$ $Case\ when\ 2 < 1\ then\ "true"\ else\ "false"\ end \rightarrow false$ $If(1<null, "true", "false") \rightarrow (Empty)$ $If(1 < 1/0, "true", "false") \rightarrow (Empty)$
$Arg1 > Arg2$	Returns true if $Arg1$ is greater than $Arg2$ . The arguments are of type Real and the result is of type Boolean. See operator < for the definition of valid arguments.
$Arg1 \leq Arg2$	Returns true if $Arg1$ is less than or equal to $Arg2$ . The arguments are of type Real and the result is of type Boolean. See operator < for the definition of valid arguments.

Operator	Description
<i>Arg1</i> >= <i>Arg2</i>	Returns true if <i>Arg1</i> is greater than or equal to <i>Arg2</i> . The arguments are of type Real and the result is of type Boolean. See operator < for the definition of valid arguments.
<i>Arg1</i> = <i>Arg2</i>	<p>Returns true if <i>Arg1</i> is equal to <i>Arg2</i>. The arguments can be of any type, but must both be of the same type. The result is of type Boolean. If any argument is null, the result is null. For arguments of type Real, see operator &lt; for the definition of valid arguments.</p> <p>Examples:</p> <pre>If(1 = 2, "true", "false" ) → false Case when 2 = 2 then "true" else "false" end → true If("Hello" = "hello", "true", "false" ) → false If("" = null, "true", "false" ) → (Empty) If(null = null, "true", "false" ) → (Empty)</pre>
<i>Arg1</i> <=> <i>Arg2</i>	<p>Returns true if the first argument is equal to the second argument or if both arguments are null. The arguments can be of any type, but must both be of the same type. The result is of type boolean.</p> <p>Example:</p> <pre>Case when [Column1] &lt;=&gt; [Column2] then 'match' else 'no match' end</pre>
<i>Arg1</i> <> <i>Arg2</i>	Returns true if <i>Arg1</i> is not equal to <i>Arg2</i> . The arguments can be of any type, but must both be of the same type. The result is of type Boolean. If any argument is invalid, the result is invalid. For arguments of type Real, see operator < for the definition of valid arguments.
<i>Arg1</i> ~= <i>Arg2</i>	<p>Operator which can be part of an IF or a CASE statement. The arguments can be of any type, but will be treated as string columns. Returns true if the <i>Arg2</i> regular expression string matches the <i>Arg1</i> string.</p> <p>Some characters, like for instance the backslash character "\", need to be escaped to work when using calculated columns. See literature about regular expression language elements, e.g., on <a href="#">MSDN</a>, for more information.</p> <p>Examples:</p> <pre>If( "aab" ~= "a+" , "true", "false" ) → true Case when "aba" ~= ".a+\$" then "true" else "false" end → true</pre>
And( <i>Arg1</i> , ...)	<p>Operator which can be part of an IF or CASE statement. It has two boolean expressions as arguments and returns true if both expressions are true.</p> <p>Examples:</p> <pre>If( 1 &lt; 2 and 2 &lt; 3, "true", "false" ) Case when false and true then "true" else "false" end</pre>
Not( <i>Arg1</i> )	<p>Operator which can be part of an IF or CASE statement. It negates the boolean expression given as argument.</p> <p>Examples:</p> <pre>If( not 1 &lt; 2, "true", "false" ) Case when not true then "true" else "false" end</pre>
Or( <i>Arg1</i> , ...)	<p>Operator which can be part of an IF or CASE statement. It has two boolean expressions as arguments and returns true if one of the expressions is true.</p> <p>Examples:</p> <pre>If( 1 &lt; 2 or 2 &lt; 3, "true", "false" ) Case when false or true then "true" else "false" end</pre>

Operator	Description
<code>Xor(Arg1, ...)</code>	<p>Can be part of an IF or CASE statement. It has two boolean expressions as arguments and returns true if exactly one of the expressions is true.</p> <p>Examples:</p> <p><code>If(1 &lt; 2 xor 2 &lt; 3, true, false)</code></p> <p><code>Case when [A]&gt;10 xor [B]&gt;5 then 1 else 0 end</code></p>

## Operator precedence

Expressions are evaluated according to the hierarchy of operators.

The table below displays the hierarchy of operators with the highest precedence operator shown first.

Expressions inside parentheses are evaluated first; nested parentheses are evaluated from the innermost parentheses to the outer.

Operators in the same row in the chart have equal precedence.

Operators	Type	Order of Evaluation
<code>()</code>	Parentheses	left to right
<code>- +</code>	Unary minus and plus	right to left
<code>* / %</code>	Multiplicative	left to right
<code>+ -</code>	Additive	left to right
<code>&amp;</code>	Concatenation	left to right
<code>&lt; &gt; &lt;= &gt;=</code>	Relational	left to right
<code>= &lt;&gt; &lt;=&gt;</code>	Equality	left to right

## Functions

A number of different functions are available for use in expressions.

The functions are grouped according to their type:

- [Binning functions](#)
- [Conversion functions](#)
- [Cast method](#)
- [Date and time functions](#)
- [Keywords in expressions](#)
- [Logical functions](#)
- [Math functions](#)
- [OVER functions](#)
- [Property functions](#)
- [Ranking functions](#)
- [Spatial functions](#)



- [Statistical functions](#)
- [Text functions](#)


## Binning functions

Binning is a way to group a number of more or less continuous values into a smaller number of "bins".

Function	Description
<code>AutoBinNumeric</code>	<p>Attempts to group the values in the specified column into the defined number of bins. However, there may be more or less bins than the defined number because the method will always result in bins with limits that are easy to read (with as even numbers as possible).</p> <p>The first argument is the column to bin and the second argument is the number of bins to aim for.</p> <p>Example:</p> <pre>AutoBinNumeric([Column], 10)</pre>

Function	Description
BinByDateTime	<p>Groups the values into bins based on a natural date, datetime or time hierarchy.</p> <p>The first argument is the Date, Time or DateTime column to bin. The second argument is the definition of the levels in the hierarchy. The hierarchy levels should be written in the form of a string containing the desired date parts, separated by dots, for example "Year.Quarter.Month". The third argument is the pruning level which specifies the level of the hierarchy to display.</p> <p>If you write a custom expression based on a BinByDateTime expression you will see a column selector with all levels of the specified hierarchy available, but with the hierarchy slider handle positioned at the specified pruning level.</p> <p>Valid arguments for Arg2 are combinations of:</p> <p>'year' or 'yy' - The year.</p> <p>'quarter' or 'qq' - The quarter.</p> <p>'month' or 'mm' - The month.</p> <p>'day of year' or 'dy' - The day of year.</p> <p>'day' or 'dd' - The day.</p> <p>'week' or 'wk' - The week.</p> <p>'day of week' or 'dw' - The weekday.</p> <p>'hour' or 'hh' - The hour.</p> <p>'minute' or 'mi' - The minute.</p> <p>'second' or 'ss' - The second.</p> <p>'millisecond' or 'ms' - The millisecond.</p> <p>Alternatively, Arg2 can also be a combination of:</p> <p>'isoyear' - The ISO 8601 week-numbering year.</p> <p>'isoweek' - The ISO 8601 week number.</p> <p>Examples:</p> <pre>BinByDateTime([Column], "Year.Quarter.Month.Day", 2)</pre> <p>For a date column called Order Date, the expression above would result in the column selector and hierarchy slider shown below, when used in a custom expression:</p>  <p>Pruning level 0 would set the slider handle to the "year" position, 1 would mean the quarter, 2 the month, and 3 the day.</p> <pre>BinByDateTime([Column], "ISOYear.ISOWeek", 1)</pre> <p>To use binning according to ISO 8601 for a date column, use the isoyear and isoweek date parts rather than the regular year and week date parts.</p>
BinByEvenDistance	<p>Groups the values in the specified column so there is an even distance between all bins.</p> <p>The first argument is the column to bin and the second is the interval length. The third (optional) argument specifies where to start counting. If no start argument is specified, the first bin starts at zero.</p> <p>Examples:</p> <pre>BinByEvenDistance([Column], 10)</pre> <pre>BinByEvenDistance([Column], 10, 5)</pre>

Function	Description
BinByEvenDistribution	<p>Groups the values in the specified column so that each bin (group) has the same number of unique values as the others. The last bin may have more unique values than the others.</p> <p>The first argument is the column to bin and the second argument is the number of bins. Invalid values will give an invalid result.</p> <p>Example:</p> <pre>BinByEvenDistribution([Column], 5) BinByEvenDistribution(Rank([Column])*Count() + RowId(), 3)</pre>
BinByEvenIntervals	<p>Groups the values in the specified column so that the value range is divided into equal intervals. The first argument is the column to bin and the second argument is the number of bins.</p> <p>Example:</p> <pre>BinByEvenIntervals([Column], 5)</pre>
BinBySpecificLimits	<p>Groups the values in the specified column by defined limits for the bins (groups). The first argument is the column to bin and the following arguments are the limits for the bins. All rows which have values larger than the largest limit will have the same bin value. Invalid values will give an invalid result.</p> <p>Example:</p> <pre>BinBySpecificLimits([Column], 1, 2, 3, 10)</pre>
BinByStdDev	<p>Groups the values into different bins depending on the values' distance from the mean, measured in standard deviations. The first argument is the column to bin and the following arguments are the number of standard deviations to show from the mean. The standard deviation arguments should be given in ascending order and all values should be positive.</p> <p>Example:</p> <pre>BinByStdDev([Column], 0.5, 1)</pre> <p>This will create a binning for:</p> <ul style="list-style-type: none"> <li>≤ -1 standard deviation</li> <li>-1 standard deviation</li> <li>-0.5 standard deviation</li> <li>0.5 standard deviation</li> <li>1 standard deviation</li> <li>&gt; 1 standard deviation</li> </ul>
BinBySubstring	<p>Groups the values into bins based on the beginning or end characters of the value. This means that you can use this method to group all values beginning with "A" into one bin, "B" into another, etc.</p> <p>The first argument is the <code>String</code> column to bin and the following is the number of characters in the substring. If the second argument is negative the substring starts from the end of the value.</p> <p>Examples:</p> <pre>BinBySubstring([Column], 1) BinBySubstring([Column], -4) BinBySubstring(String([Integer Column]), 1)</pre>

Function	Description
<code>BinByTimeSpan</code>	<p>Groups the values into bins based on a time span.</p> <p>The first argument is the <code>TimeSpan</code> column to bin. The second argument is the hierarchy level definition written in the form of a string containing the desired time span parts, separated by dots (e.g., "Hours.Minutes"). The third argument is the zero-based pruning level which specifies the level of the hierarchy to display.</p> <p>Valid arguments for <code>Arg2</code> are combinations of:</p> <ul style="list-style-type: none"> <li>'Days' - The days.</li> <li>'Hours' - The hours.</li> <li>'Minutes' - The minutes.</li> <li>'Seconds' - The seconds.</li> <li>'Milliseconds' - The milliseconds.</li> </ul> <p>Example:</p> <pre>BinByTimeSpan([Column], "Days.Hours.Minutes.Seconds", 2)</pre> <p>For a <code>TimeSpan</code> column called <code>Time Difference</code>, the expression above would result in the column selector and hierarchy slider shown below when used in a custom expression:</p> 
<code>FiscalBinByDateTime</code>	<p>Groups the values into bins based on a natural date or datetime hierarchy, with the levels shifted the specified number of months.</p> <p>The first argument is the <code>Date</code> or <code>DateTime</code> column to bin. The second argument is the hierarchy level definition written in the form of a string containing the desired date parts, separated by dots (e.g., "Year.Quarter.Month"). Only <code>Year</code>, <code>Quarter</code> and <code>Month</code> are supported. The third argument is the pruning level which specifies the level of the hierarchy to display. The fourth argument (optional) is the number of months to shift.</p> <p>If no fourth argument is specified then the value of the document property <code>FiscalYearOffset</code> will be used.</p> <p>Examples:</p> <pre>FiscalBinByDateTime([Date Column], "Year.Quarter.Month", 1, 2)</pre> <pre>FiscalBinByDateTime([Date Column], "Year.Quarter.Month", 2)</pre>



If an axis using a custom expression with binning has been set up to `Evaluate axis expression on: Current filtering only` (applicable using the Windows client only), then the binning will be recalculated with each filtering.

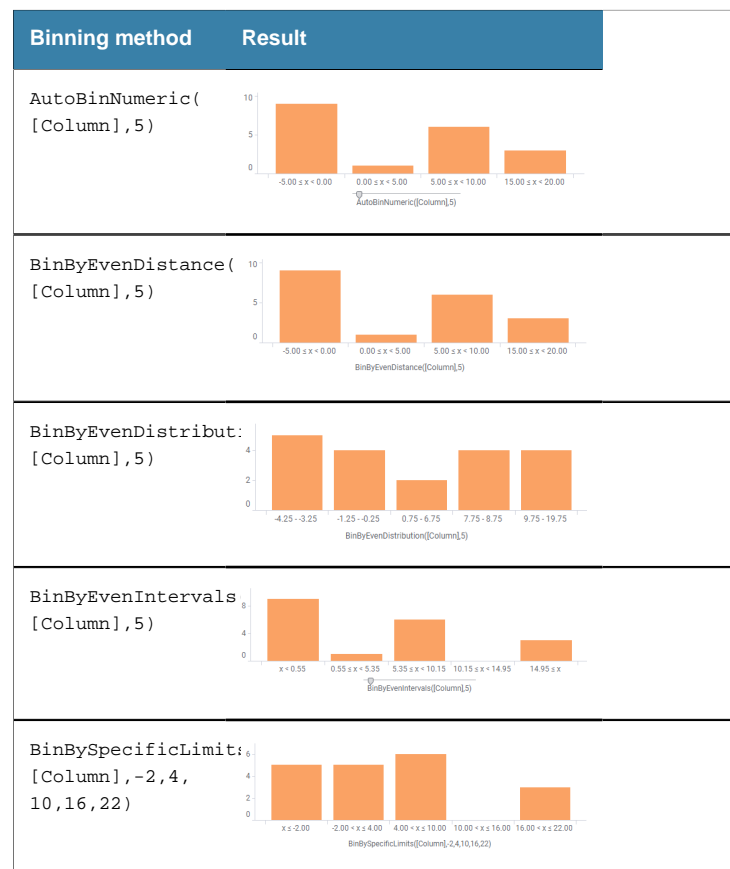
See also [Functions](#).

## Example

Consider a column with the following values:

-0.25  
8.75  
16.75  
-3.25  
0.75  
9.75  
-1.25  
8.75  
18.75  
-4.25  
-0.25  
7.75  
-3.25  
6.75  
19.75  
-4.25  
-1.25  
8.75  
-3.25

If the column is binned using the binning methods below, the resulting limits are as follows:



**Tip:** You can [change the formatting](#) of the column to view fewer (or no) decimals on the axis.

## Conversion functions

Available conversion functions are listed.


Function	Description
<code>Base64Decode(Arg1)</code>	<p>Decodes a base64 string into a binary large object so it can be viewed as an image in Spotfire.</p> <p>Example:</p> <pre>Base64Decode([Column])</pre>
<code>Base64Encode(Arg1)</code>	<p>Encodes a binary large object as a base64 string. This might be of interest if data from Spotfire is to be exported to other applications.</p> <p>Example:</p> <pre>Base64Encode([Column])</pre>
<code>Boolean(Arg1)</code>	<p>Converts the column or value to a Boolean.</p> <p>Example:</p> <pre>Boolean([Column])</pre>
<code>Cast(Arg1 as type)</code>	<p>Casts any expression to any type (except Null/Undefined).</p> <p>Invalid values are propagated. Casting performed for different types of input and output types results in different outputs. See <a href="#">Cast method</a> for more information.</p> <p>Example:</p> <pre>Cast([IntegerColumn] as Currency)</pre>
<code>Currency(Arg1)</code>	<p>Converts the column or value to a Currency.</p> <p>Example:</p> <pre>Currency([Column])</pre>

Function	Description
<code>Date(Arg1, ..., Arg3)</code>	<p>Converts the column or values to a <code>Date</code>. If a single argument is used, <i>Arg1</i> can be of type <code>String</code> or <code>DateTime</code>. If a <code>String</code> is specified, the date must be written in a format that Spotfire can recognize. Additionally, all parts of the date (year, month and day) must be present. See examples below. If a <code>DateTime</code> is specified, the time part is removed.</p> <p>If three arguments are given, they must be <code>Integer</code> values. The first argument is the year, the second is the month and the third is the day of the month.</p> <p>If a single <code>Integer</code> argument is given, the <code>Date</code> function will interpret the <code>Integer</code> as ticks. Ticks are counted with the unit 100 nanoseconds, starting from the date 01-01-01 (January first, year one).</p> <p>See also <a href="#">Date and time functions</a> and <a href="#">Cast method</a>.</p> <p>Examples:</p> <pre>Date("2003-03-21") → 3/21/2003 Date("3/21/03") → 3/21/2003 Date("10") → (Empty) Date(null) → (Empty) Date("2003-03-21 11:37:00") → 3/21/2003 Date(2003, 03, 21) → 3/21/2003 Date(6050000000000000001) → 3/4/1918</pre> <p>(The output formats available are dependent on your current locale.)</p>
<code>DateTime(Arg1, Arg2, ..., Arg7)</code>	<p>Converts the column or values to a <code>DateTime</code>. If a single argument is used, <i>Arg1</i> can be of type <code>String</code> or <code>Date</code>. If a <code>String</code> is specified, the date must be written in a format that Spotfire can recognize. Additionally, at least all parts of the date (year, month and day) must be present. If a <code>Date</code> is specified, the time part is set to 00:00:00 (12:00:00 AM).</p> <p>If seven arguments are given, they must be <code>Integer</code> values. The first argument is the year, the second is the month, the third is the day of the month, the fourth is the hour, the fifth is the minute, the sixth is the second and the seventh argument is the millisecond.</p> <p>If a single <code>Integer</code> argument is given, the <code>DateTime</code> function will interpret the integer as ticks. Ticks are counted with the unit 100 nanoseconds, starting from the date 01-01-01 (January first, year one).</p> <p>See also <a href="#">Date and time functions</a> and <a href="#">Cast method</a>.</p> <p>Examples:</p> <pre>DateTime("2003-03-21 11:37:00") → 3/21/2003 11:37:00 AM DateTime("10") → (Empty) DateTime(null) → (Empty) DateTime("2003-03-21") → 2003-03-21 00:00:00 DateTime(2003, 03, 21, 11, 37, 00) → 2003-03-21 11:37:00 DateTime(6050000000000000001) → 3/4/1918 11:33:20 AM</pre> <p>(The output formats available depend on your current locale.)</p>
<code>FirstValidAfter(Arg1)</code>	<p>Substitutes null values in a column with the first valid value after the missing value.</p> <p>Example:</p> <pre>FirstValidAfter([Column])</pre>

Function	Description
<code>FromEpochMilliseconds(Arg1)</code>	<p>Converts the column or value to a datetime, where numbers are interpreted as the number of milliseconds that have passed since 1970-01-01.</p> <p>Example:</p> <pre>FromEpochMilliseconds(231312)</pre>
<code>FromEpochSeconds(Arg1)</code>	<p>Converts the column or value to a datetime, where numbers are interpreted as the number of seconds that have passed since 1970-01-01.</p> <p>Example:</p> <pre>FromEpochSeconds(231312)</pre>
<code>Integer(Arg1)</code>	<p>Converts the column or value to an <code>Integer</code> number. If the conversion fails, an error is returned. <i>Arg1</i> can be of type <code>Integer</code>, <code>Real</code> or <code>String</code>, and the result is of type <code>Integer</code>. Real numbers are truncated, i.e., only the integer part is used.</p> <p>Examples:</p> <pre>Integer("-123") → -123 Integer(-2.99) → -2 Integer("0%") → (Empty) Integer(1e20) → (Empty) Integer(null) → (Empty)</pre>
<code>LastValidBefore(Arg1)</code>	<p>Substitutes null values in a column with the last valid value before the missing value.</p> <p>Example:</p> <pre>LastValidBefore([Column])</pre>
<code>LongInteger(Arg1)</code>	<p>Converts the column or value to a <code>LongInteger</code>.</p> <p>Example:</p> <pre>LongInteger([Column])</pre>
<code>ParseDate(Arg1, ..., Arg3)</code>	<p>Parses a date from a string to a <code>Date</code> format. The first argument is a string or a string column containing a date.</p> <p>The second argument is a <a href="#">format string</a> explaining how the date is built up and the third (optional) argument is a culture code.</p> <p>Examples:</p> <pre>ParseDate("2013-09-17", "yyyy-MM-dd") ParseDate("17-okt-2013", "dd-MMM-yyyy", "sv-SE")</pre>
<code>ParseDateTime(Arg1, ..., Arg3)</code>	<p>Parses date/time information from a string to a <code>DateTime</code> format. The first argument is a string or a string column containing date/time information.</p> <p>The second argument is a <a href="#">format string</a> explaining how the information is built up and the third (optional) argument is a culture code.</p> <p>Examples:</p> <pre>ParseDateTime("2013-09-17 10:30", "yyyy-MM-dd h:mm") ParseDateTime("17-okt-2013 13:25", "dd-MMM-yyyy HH:mm", "sv-SE")</pre>



Function	Description
<code>ParseReal(Arg1, Arg2)</code>	<p>Parses a string to a <code>Real</code> value. The first argument is a string or a string column. The second (optional) argument is a culture code.</p> <p>Examples:</p> <pre>ParseReal([Column], "en-US") ParseReal("7.25")</pre>
<code>ParseTime(Arg1, ..., Arg3)</code>	<p>Parses a time from a string to a <code>Time</code> format. The first argument is a string or a string column containing a time.</p> <p>The second argument is a <a href="#">format string</a> explaining how the time is built up and the third (optional) argument is a culture code.</p> <p>Examples:</p> <pre>ParseTime("9:25", "h:mm") ParseTime("17:30", "HH:mm", "sv-SE")</pre>
<code>ParseTimeSpan(Arg1, ..., Arg3)</code>	<p>Parses timespan information from a string to a <code>TimeSpan</code> format. The first argument is a string or a string column containing timespan information.</p> <p>The second argument is a format string explaining how the information is built up and the third (optional) argument is a culture code.</p> <p>Examples:</p> <pre>ParseTimeSpan("5.7:11:3.1", "d.h:m:s.f", "sv-SE") ParseTimeSpan("123 3 11 7 -5", "f s m h d")</pre>
<code>Real(Arg1)</code>	<p>Converts the column or value to a <code>Real</code> number. If the conversion fails, an error is returned. <i>Arg1</i> can be of type <code>Integer</code>, <code>Real</code> or <code>String</code>, and the result is of type <code>Real</code>.</p> <p>Examples:</p> <pre>Real(1.23) → 1.23 Real(2) → 2 Real("0%") → (Empty) Real(null) → (Empty)</pre>
<code>SingleReal(Arg1)</code>	<p>Converts the column or value to a <code>SingleReal</code>.</p> <p>Example:</p> <pre>SingleReal([Column])</pre>

Function	Description
<code>SN(Arg1, Arg2)</code>	<p>Substitutes null values. Returns <i>Arg1</i> if it is not null, <i>Arg2</i> otherwise. <i>Arg1</i> and <i>Arg2</i> can be of any type, but both must be of the same type or null. The result is of the same type as the arguments.</p> <p>A common usage is to substitute null values in a column. If <i>Arg1</i> is a column, <i>Arg2</i> can be either a value of the same type as the contents of the column or a column with the same content type. If <i>Arg2</i> is also a column, the error in <i>Arg1</i> will be replaced with the value from the same row in <i>Arg2</i>.</p> <p>Examples:</p> <p><code>SN(1, 2) → 1</code></p> <p><code>SN(null, 2) → 2</code></p> <p><code>SN(0/0, 2) → #NA</code></p> <p><code>SN([Column], 1) → 1</code> (if null value in column)</p> <p><code>SN([Column1], [Column2]) → (value from Column2 if null value in Column1)</code></p>
<code>String(Arg1)</code>	<p>Converts the column or value to a <i>String</i>. <i>Arg1</i> can be of any type and the result is of type string.</p> <div>  <p>This function always uses the default value formatting. This might lead to lost precision when converting numerical values.</p> </div> <p>Examples:</p> <p><code>String(1.23) → "1.23"</code></p> <p><code>String(null) → (Empty)</code></p>
<code>Time(Arg1, Arg2, ..., Arg4)</code>	<p>Converts the column or values to a <i>Time</i>. If the conversion fails, an error is returned. If a single argument is used, <i>Arg1</i> can be of type <i>String</i> or <i>DateTime</i>. If a <i>String</i> is specified, the time must be written in a format that Spotfire can recognize. Additionally, both the hour and the minute must be specified. See examples below. If a <i>DateTime</i> is specified, the date part is removed.</p> <p>If four arguments are given, they must be <i>Integer</i> values. The first argument is the hour, the second is the minute, the third is the second and the fourth is the millisecond.</p> <p>If a single <i>Integer</i> argument is given, the <i>Time</i> function will interpret the integer as ticks. Ticks are counted with the unit 100 nanoseconds, starting from the date 01-01-01 (January first, year one).</p> <p>See also <a href="#">Date and time functions</a> and <a href="#">Cast method</a>.</p> <p>Examples:</p> <p><code>Time("11:37:00") → 11:37:00</code></p> <p><code>Time("10") → (Empty)</code></p> <p><code>Time(null) → (Empty)</code></p> <p><code>Time(11, 30, 20, 4) → 11:30:20</code></p> <p><code>Time(605000000000000001) → 11:33:20 AM</code></p> <p>(The output formats available depend on your current locale.)</p>

Function	Description
<code>TimeSpan(Arg1, Arg2, ..., Arg5)</code>	<p>Creates a <code>TimeSpan</code> from a column or values. If a single argument is given, the input column can be of type <code>String</code>, <code>TimeSpan</code>, or <code>Integer</code>. If a <code>String</code> is specified, the timespan must be written in the format "[<i>-</i>]d.h:m:s.ms".</p> <p>If five arguments are given, then the first argument is the days, the second is the hours, the third is the minutes, the fourth is the seconds and the fifth is the milliseconds. The first four arguments are <code>Integers</code>, the last is a <code>Real</code> number.</p> <p>If a single <code>Integer</code> argument is given, the <code>TimeSpan</code> will interpret the <code>Integer</code> as ticks. Ticks are counted with the unit 100 nanoseconds. That means that the <code>TimeSpan</code> will be produced as fractions of seconds using tenths of microseconds (ticks), 7 decimals, as base. If 10000 is given, the <code>TimeSpan</code> is 0.0:0:0.001.</p> <p>Examples:</p> <pre>TimeSpan([Column]) TimeSpan("247.5:17:11.5002") TimeSpan(247, 05, 17, 11, 500.2)</pre>
<code>ToEpochMilliseconds([Arg1])</code>	<p>Converts the column or value from a datetime to a number representing the number of milliseconds that have passed since 1970-01-01.</p> <p>Example:</p> <pre>ToEpochMilliseconds([Date])</pre>
<code>ToEpochSeconds([Arg1])</code>	<p>Converts the column or value from a datetime to a number representing the number of seconds that have passed since 1970-01-01.</p> <p>Example:</p> <pre>ToEpochSeconds([Date])</pre>

For a list of data types available for conversion, see [Data types](#).

See also [Functions](#).

## Cast method

You find an overview below of what will happen when a column is cast from one data type to another.

The cast method can be used to cast an argument to any data type, similar to the regular [conversion functions](#).

For example:

```
cast [Column 1] as LongInteger
```

```
cast (power(2,2) as integer)
```

Input	Output										
	Integer	Real	Currency/ Decimal	Date/ DateTime/ Time	String	Binary	LongInteger	Single	Real	TimeSpan	Boolean

Input	Output									
<b>Integer</b>	Same value.	Value cast to Real.	Value converted to Decimal if it fits in the limit. Null otherwise.	Date, DateTime, or Time value is created using the integer value as ticks*.	Formatted using input formatter.	Null.	Value cast to LongInteger.	Value converted to SingleReal.	The integer value is interpreted as ticks* and the ticks are converted to a TimeSpan.	If the value is 1 then true. If the value is 0 then false. Null otherwise.
<b>Real</b>	Integer part of real value is it fits in the limit. Null otherwise.	Same value.	Value converted to Decimal if it fits in the limit. Null otherwise.	Date, DateTime, or Time value is created using the integer part of the value as ticks*.	Formatted using input formatter.	Null.	Value cast to LongInteger if it fits in the limit. Null otherwise.	Value converted to SingleReal if it fits in the limit. Null otherwise.	The integer part is interpreted as ticks* and the ticks are converted to a TimeSpan.	If the value is 1 then true. If the value is 0 then false. Null otherwise.
<b>Currency/Decimal</b>	Integer part of decimal value is it fits in the limit. Null otherwise.	Decimal value rounded to Real if it fits, null otherwise.	Same value.	Date, DateTime, or Time value is created using the integer part of the value as ticks*.	Formatted using input formatter.	Null.	Integer part of decimal value if it fits in the limit. Null otherwise.	Decimal value rounded to SingleReal if it fits, null otherwise.	The integer part is interpreted as ticks* and the ticks are converted to a TimeSpan.	If the value is 1 then true. If the value is 0 then false. Null otherwise.
<b>Date/DateTime/Time</b>	Number of ticks* converted to Integer if it fits in the limit. Null otherwise.	Number of ticks* converted to Real.	Number of ticks* converted to Decimal if it fits in the limit. Null otherwise.	Same value.	Formatted using input formatter.	Null.	Number of ticks* converted to LongInteger if it fits in the limit. Null otherwise.	Number of ticks* converted to SingleReal if it fits in the limit. Null otherwise.	Takes ticks* and converts them to a TimeSpan.	If ticks* == 0, then false. True otherwise.

Input	Output									
<b>String</b>	Try to parse using output formatter. Null if it failed to parse.	Try to parse using output formatter. Null if it failed to parse.	Try to parse using output formatter. Null if it failed to parse.	Try to parse using output formatter. The string is interpreted as number of ticks*. Null if it failed to parse.	Same value.	Null.	Try to parse using output formatter. Null if it failed to parse.	Try to parse using output formatter. Null if it failed to parse.	Try to parse using output formatter. Null if it failed to parse.	Try to parse using output formatter. Null if it failed to parse.
<b>Binary</b>	Null.	Null.	Null.	Null.	Null.	Same value.	Null.	Null.	Null.	Null.
<b>LongInteger</b>	Value converted to Integer if it fits in the limit. Null otherwise.	Value cast to Real.	Value converted to Decimal if it fits in the limit. Null otherwise.	Date, DateTime or Time value is created using the integer value as ticks*.	Formatted using input formatter.	Null.	Same value.	Value converted to SingleReal.	The integer value is interpreted as ticks* and the ticks are converted to a TimeSpan.	If the value is 1 then true. If the value is 0 then false. Null otherwise.
<b>SingleReal</b>	Integer part of SingleReal value if it fits in the limit. Null otherwise.	Value cast to Real.	Value converted to Decimal if it fits in the limit. Null otherwise.	Date, DateTime, or Time value is created using the integer part of the value as ticks*.	Formatted using input formatter.	Null.	Value cast to LongInteger if it fits in the limit. Null otherwise.	Same value.	The integer value is interpreted as ticks* and the ticks are converted to a TimeSpan.	If the value is 1 then true. If the value is 0 then false. Null otherwise.
<b>TimeSpan</b>	Number of ticks converted to Integer if it fits in the limit. Null otherwise.	Number of ticks converted to Real if it fits in the limit. Null otherwise.	Value converted to Decimal if it fits in the limit. Null otherwise.	Takes ticks and converts them to date value.	Formatted using input formatter.	Null.	Takes ticks and converts them to LongInteger value.	Takes ticks and converts them to SingleReal value.	Same value.	If the value is 1 then true. If the value is 0 then false. Null otherwise.

Input	Output									
<b>Boolean</b>	1 if true. 0 if false.	1 if true. 0 if false.	1 if true. 0 if false.	Converts to date value using the ticks 0 and 1.	Formatted using input formatter.	Null.	1 if true. 0 if false.	1 if true. 0 if false.	Converts to TimeSpan ticks using the ticks 0 and 1.	Same value.

\* Ticks are counted with the unit 100 nanoseconds, starting from the date 01-01-01 (January first, year one).

See also [Functions](#).

## Date and time functions

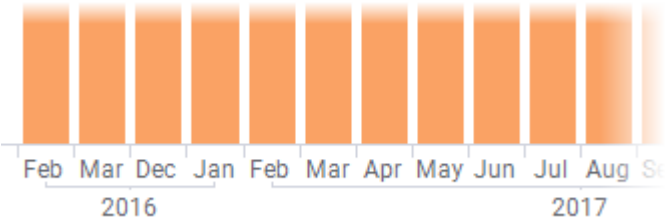
The list below shows date and time functions that can be used in expressions.

Function	Description
<code>DateAdd(Arg1, Arg2, Arg3)</code>	<p>Adds an interval to a Date, Time or a DateTime. The method can add either a TimeSpan or an Integer representing a specified date or time part (e.g., a number of days).</p> <p>If a TimeSpan is to be added, two arguments are needed: a DateTime column and a TimeSpan column.</p> <p>If an integer value is to be added to a date or time part, three arguments are used: <i>Arg1</i> is a string describing which part to add. <i>Arg2</i> is a number which contains the number of parts to add. <i>Arg3</i> is the Date, Time or DateTime column.</p> <p>Valid arguments for <i>Arg1</i> are:</p> <ul style="list-style-type: none"> <li>'year' or 'yy' - The year.</li> <li>'quarter' or 'qq' - The quarter.</li> <li>'month' or 'mm' - The month.</li> <li>'day' or 'dd' - The day.</li> <li>'week' or 'wk' - The week.</li> <li>'hour' or 'hh' - The hour.</li> <li>'minute' or 'mi' - The minute.</li> <li>'second' or 'ss' - The second.</li> <li>'millisecond' or 'ms' - The millisecond.</li> <li>'isoyear' - The ISO 8601 week-numbering year.</li> <li>'isoweek' - The ISO 8601 week number.</li> </ul> <p>Examples:</p> <pre>DateAdd([Date Column], [TimeSpan Column]) DateAdd('year', 2, [Date Column]) DateAdd('month', 1, [Date Column])</pre>

Function	Description
<code>DateDiff(Arg1, Arg2, Arg3)</code>	<p>Calculates the difference between two Date, Time or DateTime columns. The result is presented either as a TimeSpan or as a real value representing a specified time part (e.g., number of days).</p> <p>If two arguments are used (a start date column and a stop date column) the result will be a TimeSpan value displaying the total difference.</p> <p>If three arguments are used, the first argument should be the part to compare. The second argument is the start date column and the third argument is the stop date column. The result of the operation is a real value.</p> <p>Valid arguments for <i>Arg1</i> are:</p> <p>'year' or 'yy' - The year.</p> <p>'quarter' or 'qq' - The quarter.</p> <p>'month' or 'mm' - The month.</p> <p>'day of year', 'dayofyear' or 'dy' - The day of year.</p> <p>'day', 'day of month', 'dayofmonth' or 'dd' - The day.</p> <p>'year and week', 'yearandweek' or 'yywk' - The year and week.</p> <p>'week' or 'wk' - The week.</p> <p>'day of week', 'dayofweek', 'weekday' or 'dw' - The weekday.</p> <p>'hour' or 'hh' - The hour.</p> <p>'minute' or 'mi' - The minute.</p> <p>'second' or 'ss' - The second.</p> <p>'millisecond' or 'ms' - The millisecond.</p> <p>'isoyear' - The ISO 8601 week-numbering year.</p> <p>'isoweek' - The ISO 8601 week number.</p> <p>Example:</p> <pre>DateDiff([Order Date], [Delivery Date])</pre> <pre>DateDiff('day', [Order Date], [Delivery Date])</pre> <p>For examples, see <a href="#">Calculations of time differences</a> on page 736 and <a href="#">Calculating time periods</a> on page 752.</p>

Function	Description
<code>DatePart(Arg1, Arg2)</code>	<p>Returns a specified part of a Date, Time or DateTime. <i>Arg1</i> is a string describing which part of the date to get and <i>Arg2</i> is the Date, Time or DateTime column. The underlying data of the calculation is an Integer, but regional settings determine the formatted output for some of the arguments ('mm', 'qq', 'dw').</p> <p>Valid arguments for <i>Arg1</i> are:</p> <p>'year' or 'yy' - The year.</p> <p>'quarter' or 'qq' - The quarter.</p> <p>'month' or 'mm' - The month.</p> <p>'day of year', 'dayofyear' or 'dy' - The day of year.</p> <p>'day', 'day of month', 'dayofmonth' or 'dd' - The day.</p> <p>'year and week', 'yearandweek' or 'yywk' - The year and week.</p> <p>'week' or 'wk' - The week.</p> <p>'day of week', 'dayofweek', 'weekday' or 'dw' - The weekday.</p> <p>'hour' or 'hh' - The hour.</p> <p>'minute' or 'mi' - The minute.</p> <p>'second' or 'ss' - The second.</p> <p>'millisecond' or 'ms' - The millisecond.</p> <p>'isoyear' - The ISO 8601 week-numbering year.</p> <p>'isoweek' - The ISO 8601 week number.</p> <p>Example:</p> <p><code>DatePart('year', [Date Column])</code></p>
<code>DateTimeNow()</code>	<p>Returns the current system time.</p> <p>Note that when running the expression in a web client or using Automation Services, it is the time in the time zone of the service (on the server) that is returned.</p> <p>Example:</p> <p><code>DateTimeNow()</code></p> <p>For examples, see <a href="#">Calculations of time differences</a> on page 736 and <a href="#">Calculating time periods</a> on page 752.</p>
<code>Day(Arg1)</code>	<p>Extracts the day of month from a Date or DateTime column. The result is an integer between 1 and 31.</p> <p>Example:</p> <p><code>Day([Date Column])</code></p>
<code>DayOfMonth(Arg1)</code>	<p>Extracts the day of month from a Date or DateTime column. The result is an integer between 1 and 31.</p> <p>Example:</p> <p><code>DayOfMonth([Date Column])</code></p>
<code>DayOfWeek(Arg1)</code>	<p>Extracts the day of week from a Date or DateTime column. The underlying data of the new column is an integer between 0 (Sunday) and 6 (Saturday), but regional settings determine the start of the week as well as the formatted output.</p> <p>Example:</p> <p><code>DayOfWeek([Date Column])</code></p>



Function	Description
<code>DayOfYear (Arg1)</code>	<p>Extracts the day of year for a Date or DateTime column. Returns an integer between 1 and 366.</p> <p>Example:</p> <pre>DayOfYear([Date Column])</pre>
<code>Days (Arg1)</code>	<p>Returns the number of days for a TimeSpan as an Integer value between -10675199 and 10675199;</p> <p>or, if the argument is an Integer value, creates a TimeSpan column where the input is the number of days.</p> <p>Examples:</p> <pre>Days([TimeSpan Column])</pre> <pre>Days(3)</pre>
<code>FiscalMonth(Arg1, Arg2)</code>	<p>Returns the fiscal month for a date.</p> <p>The first argument is the Date column. The second argument (optional) is the number of months to shift.</p> <p>If no second argument is specified then the value of the document property <code>FiscalYearOffset</code> will be used. Regional settings determine the formatted output.</p> <p>When used in hierarchies, the <code>FiscalMonth</code> method keeps track of the sort order of the months so that visualization items are shown in the correct order. For example, if the fiscal year ends with January, then the next fiscal year should start with February as shown below and not with January:</p>  <p>The hierarchy in the image is built using the following expression:</p> <pre>&lt;FiscalYear([Date Column],1) NEST FiscalMonth([Date Column], 1)&gt;</pre> <p>Examples:</p> <pre>FiscalMonth([Date Column], 1)</pre> <pre>FiscalMonth([Date Column])</pre>
<code>FiscalQuarter (Arg1, Arg2)</code>	<p>Returns the fiscal quarter for a date (<i>Arg1</i>), based on a shift specified in months (<i>Arg2</i>).</p> <p>The second argument is optional. If no second argument is specified then the value of the document property <code>FiscalYearOffset</code> will be used.</p> <p>Examples:</p> <pre>FiscalQuarter([Date Column], 1)</pre> <pre>FiscalQuarter([Date Column])</pre>

Function	Description
<code>FiscalYear (Arg1, Arg2)</code>	<p>Returns the fiscal year for a date (<i>Arg1</i>), based on a shift specified in months (<i>Arg2</i>).</p> <p>The second argument is optional. If no second argument is specified then the value of the document property <code>FiscalYearOffset</code> will be used.</p> <p>Examples:</p> <pre>FiscalYear([Date Column], 1) FiscalYear([Date Column])</pre>
<code>Hour(Arg1)</code>	<p>Extracts the hour from a <code>DateTime</code> or <code>Time</code> column. Returns an integer between 0 and 23.</p> <p>Example:</p> <pre>Hour([Time Column])</pre>
<code>Hours(Arg1)</code>	<p>Returns the number of hours for a <code>TimeSpan</code> as an <code>Integer</code> value between 0 and 23;</p> <p>or, if the argument is an <code>Integer</code> value, creates a <code>TimeSpan</code> column where the input is the number of hours.</p> <p>Examples:</p> <pre>Hours([TimeSpan Column]) Hours(10)</pre>
<code>ISOWeek(Arg1)</code>	<p>Returns the ISO week according to ISO 8601 for a date (<code>Date</code> or <code>DateTime</code>) as an integer between 1 and 53.</p> <p>Example:</p> <pre>ISOWeek([Date Column])</pre>
<code>ISOYear(Arg1)</code>	<p>Returns the ISO week-numbering year according to ISO 8601 from a date (<code>Date</code> or <code>DateTime</code>) as an integer.</p> <p>Example:</p> <pre>ISOYear([Date Column])</pre>
<code>Millisecond(Arg1)</code>	<p>Extracts the millisecond from a <code>DateTime</code> or <code>Time</code> column. Returns an integer between 0 and 999.</p> <p>Example:</p> <pre>Millisecond([Time Column])</pre>
<code>Milliseconds(Arg1)</code>	<p>Returns the number of milliseconds for a <code>TimeSpan</code> as an <code>Integer</code> value between 0 and 999;</p> <p>or, if the argument is a <code>Real</code> value, creates a <code>TimeSpan</code> column where the input is the number of milliseconds.</p> <p>Examples:</p> <pre>Milliseconds([TimeSpan Column]) Milliseconds(123.45)</pre>
<code>Minute(Arg1)</code>	<p>Extracts the minute from a <code>DateTime</code> or <code>Time</code> column. Returns an integer between 0 and 59.</p> <p>Example:</p> <pre>Minute([Time Column])</pre>

Function	Description
<code>Minutes(Arg1)</code>	<p>Returns the number of minutes for a <code>TimeSpan</code> as an Integer value between 0 and 59;</p> <p>or, if the argument is an Integer value, creates a <code>TimeSpan</code> column where the input is the number of minutes.</p> <p>Examples:</p> <pre>Minutes([TimeSpan Column]) Minutes(50)</pre>
<code>Month(Arg1)</code>	<p>Extracts the month from a <code>Date</code> or <code>DateTime</code> column. The underlying data of the new column is an integer between 1 and 12, but regional settings determine the formatted output.</p> <p>Example:</p> <pre>Month([Date Column])</pre>
<code>Quarter(Arg1)</code>	<p>Extracts the quarter from a <code>Date</code> or <code>DateTime</code> column. The underlying data of the new column is an integer between 1 and 4, but regional settings determine the formatted output.</p> <p>Example:</p> <pre>Quarter([Date Column])</pre>
<code>Second(Arg1)</code>	<p>Extracts the second from a <code>DateTime</code> or <code>Time</code> column. Returns an integer between 0 and 59.</p> <p>Example:</p> <pre>Second([Time Column])</pre>
<code>Seconds(Arg1)</code>	<p>Returns the number of seconds for a <code>TimeSpan</code> as an Integer value between 0 and 59;</p> <p>or, if the argument is an Integer value, creates a <code>TimeSpan</code> column where the input is the number of seconds.</p> <p>Examples:</p> <pre>Seconds([TimeSpan Column]) Seconds(10)</pre>
<code>Today()</code>	<p>Returns the current date.</p> <p>Note that when running the expression in a web client or using Automation Services, it is the date in the time zone of the service (on the server) that is returned.</p> <p>Example:</p> <pre>Today()</pre>
<code>TotalDays(Arg1)</code>	<p>Returns the number of days for a <code>TimeSpan</code> as a real value expressed in whole days and fractional days.</p> <p>Example:</p> <pre>TotalDays([TimeSpan Column])</pre>
<code>TotalHours(Arg1)</code>	<p>Returns the number of hours for a <code>TimeSpan</code> as a real value expressed in whole and fractional hours.</p> <p>Example:</p> <pre>TotalHours([TimeSpan Column])</pre>

Function	Description
<code>TotalMilliseconds(Arg1)</code>	<p>Returns the number of milliseconds for a <code>TimeSpan</code> as a real value expressed in whole and fractional milliseconds.</p> <p>Example:</p> <pre>TotalMilliseconds([TimeSpan Column])</pre>
<code>TotalMinutes(Arg1)</code>	<p>Returns the number of minutes for a <code>TimeSpan</code> as a real value expressed in whole and fractional minutes.</p> <p>Example:</p> <pre>TotalMinutes([TimeSpan Column])</pre>
<code>TotalSeconds()</code>	<p>Returns the number of seconds for a <code>TimeSpan</code> as a real value expressed in whole and fractional seconds.</p> <p>Example:</p> <pre>TotalSeconds([TimeSpan Column])</pre>
<code>Week(Arg1)</code>	<p>Extracts the week from a <code>Date</code> or <code>DateTime</code> column as an integer between 1 and 54, where the first week of year is dependent on the regional settings. See also <a href="#">ISOWeek</a>.</p> <p>Example:</p> <pre>Week([Date Column])</pre>
<code>Year(Arg1)</code>	<p>Extracts the year from a <code>Date</code> or <code>DateTime</code> column. The result is of type <code>Integer</code>. See also <a href="#">ISOYear</a>.</p> <p>Example:</p> <pre>Year([Date Column])</pre>
<code>YearAndWeek(Arg1)</code>	<p>Extracts the year and week from a <code>Date</code> or <code>DateTime</code> column. Returns an <code>Integer</code> (<code>Year*100 + Week number</code>), for example, the date 2005-10-13 will return 200541. The calculation depends on regional settings, and might produce undesired results when crossing year boundaries. See also <a href="#">ISOYear</a>, <a href="#">ISOWeek</a>, and <a href="#">BinByDateTime</a>.</p> <p>Example:</p> <pre>YearAndWeek([Date Column])</pre> <p>For more information, see <a href="#">Additional operations with time hierarchies</a> on page 510.</p>

See also [Functions](#).

## Keywords in expressions

There are a number of different keywords that can be included in different types of expressions. See below for a description. The keywords are not case-sensitive.

Option	Description
AS	<p>The AS keyword can be used to specify the name of an expression in the context of a custom expression.</p> <p>Examples:</p> <pre>Sum([Sales]) as [Total Sales]</pre> <pre>&lt;[Region_Name] as [Region]&gt;</pre>

Option	Description
CROSS	<p>The CROSS keyword can be used in categorical custom expressions to specify that a cross join should be performed between the category levels. This means that all possible combinations of categories are displayed, even those that currently hold no data.</p> <p>Example:</p> <pre>&lt;[A] CROSS [B]&gt;</pre>
DISTINCT	<p>The DISTINCT keyword can be used within aggregated expressions to specify that the aggregation should be computed on the distinct values only.</p> <p>Example:</p> <pre>Avg(distinct [Column])</pre>
ELSE	<p>ELSE is used in case statements to define what happens if a condition is not met.</p> <p>The case statement has two different forms.</p> <p>Simple:</p> <pre>case Arg1 when Arg2 then Arg3 else Arg4 end</pre> <p>The <i>Arg1</i> expression is evaluated and when <i>Arg1</i> is equal to <i>Arg2</i> then <i>Arg3</i> is returned.</p> <p>Multiple when/then expressions can be entered and are evaluated in left to right order.</p> <p>Searched:</p> <pre>case when Arg1 then Arg2 else Arg3 end</pre> <p>Returns <i>Arg2</i> if <i>Arg1</i>=true, and <i>Arg3</i> if <i>Arg1</i>=false.</p> <p>Multiple when/then expressions can be entered and are evaluated in left to right order.</p> <p>Examples:</p> <pre>Case when 1 &lt; 2 then "a" when 1 &lt; 3 then "b" else "c" end Case [Column] when 3 then "a" when 2 then "b" else "c" end</pre>
END	<p>END is used in case statements to signal the end of the statement.</p> <p>See ELSE for examples of case statements.</p>
FALSE	<p>FALSE is a boolean constant that can be used in logical expressions.</p> <p>Example:</p> <pre>false</pre>
IS	<p>IS is a keyword that can be used check whether an expression is null or not.</p> <p>Examples:</p> <pre>[Column] Is Not Null Avg([Column]) Is Null</pre>
NEST	<p>The NEST keyword can be used in categorical custom expressions to specify that the category levels should be nested. This means that only those combinations of values that actually hold data are shown.</p> <p>Example:</p> <pre>&lt;[A] nest [B]&gt;</pre>

Option	Description
NULL	<p>NULL is a keyword that can either be used to specify a null (empty) value, or, it can be a part of the 'Is Not Null' or 'Is Null' operators.</p> <p>Examples:</p> <pre> null  [Column] Is Not Null </pre>
OVER	<p>OVER is a keyword that can be used after an aggregation to specify that the aggregation should be calculated over another window than the default. The OVER keyword is always used together with an OVER function which specifies how to group the data, based on the nodes in the referenced hierarchy.</p> <p>Example:</p> <pre> Sum([Sales]) OVER (AllPrevious([Axis.X])) </pre>
THEN	<p>The THEN keyword can be used in two different contexts: If it is part of a CASE expression, it simply determines what happens when a condition is met. THEN can also be used to calculate post-aggregation expressions, in expressions on continuous axes.</p> <p>In the post-aggregation context, the part of the expression after the THEN keyword is calculated on top of the aggregated data, and the aggregated value is referred to using the [Value] name.</p> <p>Examples:</p> <pre> CASE WHEN 1 &lt; 2 THEN "a" WHEN 1 &lt; 3 THEN "b" ELSE "c" end  Sum([Sales]) THEN Sum([Value]) OVER (AllPrevious([Axis.X])) </pre>
TRUE	<p>TRUE is a boolean constant that can be used in logical expressions.</p> <p>Example:</p> <pre> true </pre>
WHEN	<p>WHEN is used in case statements to define what happens if a condition is met.</p> <p>See ELSE for examples of case statements.</p>

See also [Functions](#).

## Logical functions

The list below shows the logical functions that can be used in expressions.

The logical functions offer even more flexibility in the expression language. For example, you can use `if` or `case` for simple conditions:

```
case [Column1] when 10 then 'ten' else 'not ten' end
```

```
case when Column1 is not null then Column1 else Column2 end
```

See below for more examples.

Function	Description
Case	<p>The case statement has two different forms.</p> <p><b>Simple:</b></p> <pre>case Arg1 when Arg2 then Arg3 else Arg4 end</pre> <p>The Arg1 expression is evaluated and when Arg1 is equal to Arg2 then Arg3 is returned. Multiple when/then expressions can be entered and are evaluated in left to right order.</p> <p><b>Searched:</b></p> <pre>case when Arg1 then Arg2 else Arg3 end</pre> <p>Returns Arg2 if Arg1=true, and Arg3 if Arg1=false. Multiple when/then expressions can be entered and are evaluated in left to right order.</p> <p>See below for more information.</p> <p><b>Example:</b></p> <pre>case when 1 &lt; 2 then "a" when 1 &lt; 3 then "b" else "c" end case [Column] when 3 then "a" when 2 then "b" else "c" end</pre> <p>For more examples, see below and <a href="#">Calculated columns based on conditions</a> on page 739.</p>
If (Arg1, Arg2, Arg3)	<p>Returns Arg2 if Arg1=true, and Arg3 if Arg1=false. Arg1 is of type boolean, usually the result of a comparison. Arg2 and Arg3 can be of any type, but must both be of the same type or null.</p> <p>The second and third argument only process subsets of the rows, which impacts all column-based and accumulating methods. See below for more information.</p> <p><b>Examples:</b></p> <pre>If([Count] &gt; 3, "many", "few") If(true, null, null) -&gt; (Empty) If(true, 1, null) -&gt; 1 If(false, null, 2) -&gt; 2 If(null, 1, 2) -&gt; (Empty) If(1 &lt; 2, "Small", "Big") -&gt; Small If([Column] Is Null, "0", "has value")</pre> <p>For more examples, see below and <a href="#">Calculated columns based on conditions</a> on page 739.</p>

Function	Description
Is Not Null	<p>Used within an If- or Case- statement, to determine whether or not an expression yields an empty value (null value).</p> <p>Example:</p> <pre>If([Column] Is Not Null, "value was not null", "value was null")</pre> <p>If an expression contains empty values (null values), you can use the SN function (see <a href="#">Conversion Functions</a>) to substitute the null values with the specified value.</p>
Is Null	<p>Used within an If- or Case- statement, to determine whether or not an expression yields an empty value (null value).</p> <p>Example:</p> <pre>If([Column] Is Null, "value was null", "value was not null")</pre> <p>If an expression contains empty values (null values), you can use the SN function (see <a href="#">Conversion Functions</a>) to substitute the null values with the specified value.</p>

### More about using if in expressions:

When you are using an If-statement in an expression, the condition in the first argument limits the values to be evaluated in the third argument. To understand how it works, first take a simple integer column as an example:

Col1
1
2
3

Insert a calculated column using the following expression:

```
if(Col1 < 2, Max(Col1), Max(Col1))
```

This will result in a new column with two different values; one for case when the max for all values that are smaller than 2 and one showing the max value for the rest of the values (which also is the max of the entire column):

Col1	if(Col1 < 2, Max(Col1), Max(Col1))
1	1
2	3
3	3

The If-function is designed to split the column into two groups and then to continue evaluating the expression on the new groups.



If you are using a `Percentile` expression instead, you will get different results on the same percentile expression depending on how the first condition has split the column:

Col2
1
2
3
4
5
6
7
8
9
10

Insert two columns using the following two expressions:

```
if (Col2 < 4, 0, Percentile(Col2, 70))
```

```
if (Col2 < 6, 0, Percentile(Col2, 70))
```

Then the following result is obtained:

Col2	<code>if (Col2 &lt; 4, 0, Percentile(Col2, 70))</code>	<code>if (Col2 &lt; 6, 0, Percentile(Col2, 70))</code>
1	0	0
2	0	0
3	0	0
4	8.2	0
5	8.2	0
6	8.2	8.8
7	8.2	8.8
8	8.2	8.8
9	8.2	8.8
10	8.2	8.8

As you can see, the value of the percentile will change depending on how 'If' has grouped the data.

### More about using case in expressions:

When working with case expressions, the condition after "when" is calculated on the entire column. The argument after "then" is calculated on those rows defined by the "when"-condition. If multiple "when-then" pairs are included in the expression a "then" is limited by all previous conditions as well as the last one.

An example using a simple integer column:

Col1
1
2
3
4
5
6
7
8

Insert two calculated columns, called Case Min and Case Max, using the following expressions:

#### Case Min:

```
case
  when [Col1]<Avg([Col1]) then Min([Col1])
  when [Col1]<(Avg([Col1]) + 1) then Min([Col1])
  else Min([Col1])
end
```

#### Case Max:

```
case
  when [Col1] < Avg([Col1]) then Max([Col1])
  when [Col1] < (Avg([Col1]) + 1) then Max([Col1])
  else Max([Col1])
end
```

The result will be:

Col1	Case Min	Case Max
1	1	4
2	1	4
3	1	4
4	1	4
5	5	5
6	6	8

7	6	8
8	6	8

See also [Functions](#).

## Math functions

The list shows the math functions you can use in expressions.

Function	Description
<code>Abs(Arg1)</code>	Returns the absolute value of <i>Arg1</i> . The argument and the result are of type <code>Real</code> .
<code>ACos(Arg1)</code>	Returns the arccosine of <i>Arg1</i> as an angle expressed in radians in the interval $[0, \pi]$ . <i>Arg1</i> must be in the interval $[-1.0, 1.0]$ , otherwise <code>#NA</code> is returned. The argument and the result are of type <code>Real</code> .
<code>ASin(Arg1)</code>	Returns the arcsine of <i>Arg1</i> as an angle expressed in radians in the interval $[-\pi/2, \pi/2]$ . <i>Arg1</i> must be in the interval $[-1.0, 1.0]$ , otherwise <code>#NA</code> is returned. The argument and the result are of type <code>Real</code> .
<code>ATan(Arg1)</code>	Returns the arctangent of <i>Arg1</i> as an angle expressed in radians in the interval $[-\pi/2, \pi/2]$ . The argument and the result are of type <code>Real</code> .
<code>ATan2(Arg1, Arg2)</code>	Returns the arctangent of two arguments (y, x) as an angle expressed in radians (in the interval $-\pi$ and $\pi$ ). <i>Arg1</i> is the y-coordinate and <i>Arg2</i> the x-coordinate. The signs of the arguments are used to determine the quadrant of the result. The arguments and the result are of type <code>Real</code> .  Example: <code>ATan2(0, -1) → 3.14159265358979</code>
<code>Ceiling(Arg1)</code>	Rounds <i>Arg1</i> up to the nearest whole number. The argument and the result are of type <code>Real</code> .  Examples: <code>Ceiling(1.01) → 2.0</code> <code>Ceiling(-1.99) → -1.0</code>
<code>Cos(Arg1)</code>	Returns the cosine of <i>Arg1</i> where <i>Arg1</i> is an angle expressed in radians. The argument and the result are of type <code>Real</code> .
<code>Exp(Arg1)</code>	Returns $e$ (2.718281...) raised to the <i>Arg1</i> power. The argument and the result are of type <code>Real</code> .
<code>Floor(Arg1)</code>	Rounds <i>Arg1</i> down to the nearest whole number. The argument and the result are of type <code>Real</code> .  Examples: <code>Floor(1.99) → 1.0</code> <code>Floor(-1.01) → -2.0</code>
<code>Ln(Arg1)</code>	Returns the natural logarithm of <i>Arg1</i> . The arguments and the result are of type <code>Real</code> . If <i>Arg1</i> is negative, the result is a <code>#NA</code> error. If <i>Arg1</i> is zero, the result is also <code>#NA</code> .

Function	Description
<code>Log(Arg1, Arg2)</code>	Returns the logarithm of <i>Arg1</i> expressed in the base specified by <i>Arg2</i> . Equivalent to $\text{Ln}(\text{Arg1}) / \text{Ln}(\text{Arg2})$ . The arguments and the result are of type <i>Real</i> . See function <i>Ln</i> for the definition of valid arguments.
<code>Log10(Arg1)</code>	Returns the 10-based logarithm of <i>Arg1</i> . Equivalent to $\text{Ln}(\text{Arg1}) / \text{Ln}(10)$ . The arguments and the result are of type <i>Real</i> . See function <i>Ln</i> for the definition of valid arguments.
<code>Mod(Arg1, Arg2)</code>	Returns the remainder of the division of <i>Arg1</i> by <i>Arg2</i> . The arguments and the result are of type <i>Real</i> . If <i>Arg2</i> is 0, the result is a #NA error.  <i>Mod(Arg1, Arg2)</i> is defined as:  $\text{Arg1} - \text{Arg2} * \text{Floor}(\text{Arg1} / \text{Arg2})$
<code>PI()</code>	Returns the numerical constant $\pi$ .  The result is of type <i>Real</i> .
<code>Power(Arg1, Arg2)</code>	Returns <i>Arg1</i> raised to the <i>Arg2</i> power. The arguments and the result are of type <i>Real</i> .  Examples:  $\text{Power}(10, 3) \rightarrow 1000$  $\text{Power}(10, -3) \rightarrow 0.001$  $\text{Power}(0, 0) \rightarrow 1$
<code>Product(Arg1, ...)</code>	Calculates the product of the values. If one argument is given, then the result is the product of the entire column. If more than one column is given, then the result is the product of each row.  The arguments and the result are of type <i>Real</i> . Null arguments are ignored and do not contribute to the product.  Examples:  $\text{Product}([\text{Column}])$  $\text{Product}(1, 2, 3) \rightarrow 6$  $\text{Product}(-1) \rightarrow -1$  $\text{Product}(1.5, -2, 3) \rightarrow -9$  $\text{Product}(1, \text{null}, 3) \rightarrow 3$  $\text{Product}(\text{null}) \rightarrow (\text{Empty})$
<code>Rand(Arg1)</code>	Returns a random real number between 0.0 and 1.0.  The integer argument is a constant seed value that is used to initialize the random number generator. It also assures that the same values are generated if the column is recalculated.  The seed value cannot be a column reference.  Example:  $\text{Rand}(147)$

Function	Description
<code>RandBetween(Arg1, Arg2, Arg3)</code>	<p>Returns a random integer number within the specified range.</p> <p>The first and the second arguments set the range for the random numbers. These arguments can be constant values or integer column references.</p> <p>The third argument is a constant seed value that is used to initialize the random number generator. It also assures that the same values are generated if the column is recalculated.</p> <p>The seed value cannot be a column reference.</p> <p>Example:</p> <pre>RandBetween(100, -100, 147) RandBetween(0, [Column 1], 147) RandBetween([Column 1], [Column 2], 37)</pre>
<code>Round(Arg1, Arg2)</code>	<p>Rounds <i>Arg1</i> to the number of decimal places specified by <i>Arg2</i>. The arguments and the result are of type <code>Real</code>, but for <i>Arg2</i>, only the integer part is used. Note that <i>Arg2</i> can be negative to round to even 10s, 100s, etc. 0.5 is rounded upwards to a number with higher magnitude (ignoring the sign).</p> <p>Examples:</p> <pre>Round(PI(), 3) → 3.142 Round(-0.5, 0) → -1 Round(25, -1) → 30</pre>
<code>Sin(Arg1)</code>	<p>Returns the sine of <i>Arg1</i> where <i>Arg1</i> is an angle expressed in radians. The argument and the result are of type <code>Real</code>.</p>
<code>Sqrt(Arg1)</code>	<p>Returns the square root of <i>Arg1</i>. The argument and the result are of type <code>Real</code>. If <i>Arg1</i> is negative, the result is a #NA error.</p>
<code>Sum(Arg1, ...)</code>	<p>Calculates the sum of the values. If one argument is given, then the result is the sum of the entire column. If more than one column is given, then the result is the sum of each row.</p> <p>Null arguments are ignored and do not contribute to the sum.</p> <p>Examples:</p> <pre>Sum([Column]) Sum(1, 2, 3) → 6 Sum(-1) → -1 Sum(1.5, -2, 3) → 2.5 Sum(1, null, 3) → 4 Sum(null) → (Empty)</pre>
<code>Tan(Arg1)</code>	<p>Returns the tangent of <i>Arg1</i> where <i>Arg1</i> is an angle expressed in radians. The argument and the result are of type <code>Real</code>.</p>

See also [Functions](#).

## OVER functions

The OVER functions are used to determine how data should be sliced, for example, relative to time periods.



The use of OVER functions is different for custom expressions and for calculated columns. In custom expressions you can include references to axes, whereas in calculated columns you can only refer to columns or fixed hierarchies.

For more information, see [Using OVER statement to reference data slices](#) on page 760 and [Using OVER statements in calculated columns](#) on page 764.

Option	Description
All	<p>Uses all the nodes in the referenced hierarchy. This can be useful when intersecting the current node with more than one hierarchy. For example, you can show the relative sales of different product categories for each month.</p> <p>Examples of custom expressions:</p> <pre>Sum([Sales]) / Sum([Sales]) OVER (Intersect(All([Axis.Color]), [Axis.X]))</pre> <pre>Sum([Sales]) / Sum([Sales]) OVER (All([Axis.X])) * 100</pre> <p>Examples of calculated columns:</p> <pre>Sum([Sales]) OVER All([Date])</pre> <pre>Max([Sales]) OVER Intersect([Category], All([Date]))</pre>
AllNext	<p>Uses all nodes, including the current, to the end of the level.</p> <p>Example of a custom expression:</p> <pre>Sum([Sales]) OVER (AllNext([Axis.X]))</pre> <p>Examples of calculated columns:</p> <pre>Sum([Sales]) OVER AllNext([Date])</pre> <pre>Max([Sales]) OVER Intersect([Category], AllNext([Date]))</pre>
AllPrevious	<p>Uses all nodes, including the current, from the start of the level. This can be used to calculate the cumulative sum.</p> <p>Examples of custom expressions:</p> <pre>Sum([Sales]) OVER (AllPrevious([Axis.X]))</pre> <pre>Sum([Sales]) OVER (Intersect(Parent([Axis.X]), AllPrevious([Axis.X])))</pre> <p>Examples of calculated columns:</p> <pre>Sum([Sales]) OVER AllPrevious([Date])</pre> <pre>Max([Sales]) OVER Intersect([Category], AllPrevious([Date]))</pre>
FirstNode	<p>Selects the first node on the current level.</p> <p>Example of a custom expression:</p> <pre>Sum([Sales]) - Sum([Sales]) OVER (FirstNode([Axis.X]))</pre> <p>Example of a calculated column:</p> <pre>Sum([Sales]) - Sum([Sales]) OVER (FirstNode([Hierarchy.TimeHierarchy]))</pre>

Option	Description
Intersect	<p>Returns the intersected rows from nodes in different hierarchies. See also <code>AllPrevious</code> and <code>All</code>.</p> <p>Example of a custom expression:</p> <pre>Intersect(Parent([Axis.X]), All([Axis.Color]), Parent([Axis.Rows]), ...)</pre> <p>Example of a calculated column:</p> <pre>Sum([Sales]) OVER Intersect([Category], AllPrevious([Date]))</pre>
LastNode	<p>Selects the last node on the current level.</p> <p>Example of a custom expression:</p> <pre>Sum([Sales]) - Sum([Sales]) OVER (LastNode([Axis.X]))</pre> <p>Example of a calculated column:</p> <pre>Sum([Sales]) - Sum([Sales]) OVER (LastNode([Hierarchy.TimeHierarchy]))</pre>
LastPeriods	<p>Includes the current node and the n - 1 previous nodes. This can be used to calculate moving averages.</p> <p>Example of a custom expression:</p> <pre>Sum([Sales]) OVER (LastPeriods(3, [Axis.X]))/3</pre> <p>Example of a calculated column:</p> <pre>Sum([Sales]) - Sum([Sales]) OVER (LastPeriods([Hierarchy.TimeHierarchy]))</pre>
NavigatePeriod	<p>Allows you to specify your own node navigation. The method has four arguments:</p> <p><i>Arg1</i> is the hierarchy to navigate.</p> <p><i>Arg2</i> is a string value specifying the level in the hierarchy that you should navigate up to.</p> <p><i>Arg3</i> is an integer specifying the number of steps to move sideways in the hierarchy at the level specified by <i>Arg1</i>.</p> <p><i>Arg4</i> is optional and determines the level in the hierarchy to move down to. This argument can be omitted in which case a navigation to the leaf level is made.</p> <p>Examples of custom expressions:</p> <pre>Sum([Sales]) OVER NavigatePeriod([Axis.X], "Year", -1)</pre> <pre>Sum([Value]) OVER NavigatePeriod([Axis.X], "Year", 0, 0)</pre> <pre>Sum([Sales]) OVER NavigatePeriod([Axis.X], 2, -1, 2)</pre> <p>Example of a calculated column:</p> <pre>Avg([Sales]) OVER (NavigatePeriod([Hierarchy.TimeHierarchy], 0, -10)) - Avg([Sales]) OVER (NavigatePeriod([Hierarchy.TimeHierarchy], 0, 10))</pre>

Option	Description
Next	<p>Compares the current node with the next node on the same level in the hierarchy. If there is no next node, that is, if the current node is the last node for the current level, the resulting subset will not contain any rows.</p> <p>Examples of custom expressions:</p> <pre>Sum([Sales]) - Sum([Sales]) OVER (Next([Axis.X]))</pre> <pre>Count() OVER Next([Axis.X], 2)</pre> <p>Example of a calculated column:</p> <pre>Max([Sales]) OVER (Intersect([Category],Next([Year])))</pre>
NextPeriod	<p>Uses the next node which has the next value on the same level as the current node. If there is no next node, that is, if the current node is the last node for the current level, the resulting subset will not contain any rows.</p> <p>Examples of custom expressions:</p> <pre>Sum([Sales]) OVER (NextPeriod([Axis.X]))</pre> <pre>Count() OVER NextPeriod([Axis.X], 2)</pre> <p>Example of a calculated column:</p> <pre>Max([Sales]) OVER (Intersect([Category],NextPeriod([Year])))</pre>
ParallelPeriod	<p>Uses the previous parallel node with the same value on the same level as the current node. For example, this can be used to compare sales results for each month with the corresponding months the previous year.</p> <p>Example of a custom expression:</p> <pre>Sum([Sales])-Sum([Sales]) OVER (ParallelPeriod([Axis.X]))</pre> <p>Example of a calculated column:</p> <pre>Sum([Sales])-Sum([Sales]) OVER (ParallelPeriod([Hierarchy.TimeHierarchy]))</pre>
Parent	<p>Uses the parent subset of the current node. If the node does not have a parent, all rows are used as the subset.</p> <p>Examples of custom expressions:</p> <pre>Sum([Sales]) / Sum([Sales]) OVER (Parent([Axis.Color]))</pre> <pre>Sum([Sales]) / Sum([Sales]) OVER (Parent([Axis.X])) * 100</pre> <pre>Avg([Sales]) OVER Parent([Axis.X])</pre> <p>Example of a calculated column:</p> <pre>Sum([Sales]) OVER (Parent([Hierarchy.TimeHierarchy]))</pre>
Previous	<p>Uses the previous node on the same level as the current node to compare the result of the current node with the previous one. If there is no previous node, that is, if the current node is the first node for the current level, the resulting subset will not contain any rows.</p> <p>Examples of custom expressions:</p> <pre>Sum([Sales]) - Sum([Sales]) OVER (Previous([Axis.X]))</pre> <pre>Count() OVER Previous([Axis.X], 2)</pre> <p>Example of a calculated column:</p> <pre>Max([Sales]) OVER (Intersect([Category],Previous([Year])))</pre>



Option	Description
PreviousPeriod	<p>Uses the previous node which has the previous value on the same level as the current node. If there is no previous node, that is, if the current node is the first node for the current level, the resulting subset will not contain any rows.</p> <p>Examples of custom expressions:</p> <pre>Sum([Sales]) OVER (PreviousPeriod([Axis.X]))</pre> <pre>Count() OVER PreviousPeriod([Axis.X], 2)</pre> <p>Example of a calculated column:</p> <pre>Max([Sales]) OVER (Intersect([Category],PreviousPeriod([Year])))</pre>

See also [Functions](#).

## Property functions

The list shows property functions that you can use in expressions.

Function	Description
\$csearch	<p>Selects a number of columns from a data table using a limiting search expression. The first argument is a data table and the second argument is a string that contains the search expression determining which column names should be returned. The function returns a list of the (unescaped) column names from the data table that match the search expression.</p> <p>Examples:</p> <pre>\$csearch([Data Table], "*")</pre> <p>→ Returns a list of all values in the data table called Data Table.</p> <pre>\$csearch([Data Table], "Col*")</pre> <p>→ Returns a list of all values in the data table called Data Table beginning with "Col", e.g., Column 1, Column 2, etc.</p>
\$esc	<p>Replaces "]" in column names with "]" and encloses the escaped column names in "[" and "]". The argument is a property value or a property function that starts with a dollar sign (\$).</p> <p>Examples:</p> <pre>\$esc({PropertyName})</pre> <p>→ Returns the property value as a column name (within [ and ]).</p> <pre>\$esc(\$csearch([Data Table], "Col*"))</pre> <p>→ Returns a list of all columns in the data table called "Data Table" beginning with "Col", e.g., [Column 1], [Column 2], etc.</p>

Function	Description
\$map	<p>Maps a list-valued property to a single string. The first argument is a template to use for each value in the list and the second argument is a specification of how the list values should be connected in the resulting expression.</p> <p>Examples:</p> <pre>\$map("sum([\${PropertyName}])", ", ", ", ")</pre> <p>→ Returns a comma separated list of the sum of the columns included in the list-valued property, e.g., <code>sum([Column 1]), sum([Column 2])</code></p> <pre>&lt;\$map("[\${PropertyName}]", " NEST")&gt;</pre> <p>→ Returns a nested categorical hierarchy using the columns included in the list-valued property, e.g., <code>&lt;[Column 1] NEST[Column 2]&gt;</code></p>
BaseRowId	<p>Returns a unique identifier for each calculated row in the visualization. This identifier is selected from identifiers calculated on the data table. This value may change when filtering or marking is performed.</p> <p>Example:</p> <pre>BaseRowId()</pre>
ColumnProperty	<p>The first argument is a column and the second argument is the column property name, presented as a string. Returns the value of the named column property from the column. The column property value cannot be a list and the column property has to exist before creating the expression.</p> <p>Example:</p> <pre>ColumnProperty([Column], "Description")</pre>
DataTableProperty	<p>Returns the value of the data table property. The argument to the method is the name of the data table property, presented as a string.</p> <p>Example:</p> <pre>DataTableProperty("Table.CreationDate")</pre>
DocumentProperty	<p>Returns the value of the document property. Document properties can be used throughout the entire document.</p> <p>The argument to the method is the name of the document property, presented as a string.</p> <p>Example:</p> <pre>DocumentProperty("Extension.NumberOfBins")</pre>
RowId	<p>Returns a unique identifier for each calculated row in the visualization. This identifier will not change when filtering or marking is performed.</p> <p>Example:</p> <pre>RowId()</pre>

See also [Functions](#).

## Ranking functions

The list shows the ranking functions that you can use in expressions.

Function	Description
DenseRank(Arg1, Arg2, Arg3...)	<p>Returns an integer value ranking of the values in the selected column. The first argument is the column to be ranked.</p> <p>An optional argument is a string determining whether to use an ascending (default) or a descending ranking. For the highest value to retrieve rank 1, use the argument "desc", for the lowest value to retrieve rank 1, use "asc".</p> <p>Ties are given the same rank value and the highest ranking number equals the number of unique values in the column.</p> <p>Additional column arguments (optional) can be used when the column should be split into separately ranked categories.</p> <p>Examples:</p> <p>DenseRank([Sales])</p> <p>DenseRank([Sales], "desc", [Region])</p> <p>For an example, see <a href="#">Ranking</a> on page 737.</p>
Rank(Arg1, Arg2, Arg3...)	<p>Returns an integer value ranking of the values in the selected column. The first argument is the column to be ranked.</p> <p>An optional argument is a string determining whether to use an ascending (default) or a descending ranking. For the highest value to retrieve rank 1, use the argument "desc", for the lowest value to retrieve rank 1, use "asc".</p> <p>Ties are given rank values depending on optional argument values:</p> <p>"ties.method=minimum" (default),</p> <p>"ties.method=maximum", or</p> <p>"ties.method=first".</p> <p>See <i>More about ranking ties</i> below for more information about the available arguments.</p> <p>Additional column arguments (optional) can be used when the column should be split into separately ranked categories.</p> <p>Examples:</p> <p>Rank([Sales])</p> <p>Rank([Sales], "desc", [Region])</p> <p>Rank([Sales], "ties.method=first")</p> <p>For an example, see <a href="#">Ranking</a> on page 737.</p>

Function	Description
RankReal(Arg1, Arg2, Arg3...)	<p>Returns a real value ranking of the values in the selected column. The first argument is the column to be ranked.</p> <p>An optional argument is a string determining whether to use an ascending (default) or a descending ranking. For the highest value to retrieve rank 1, use the argument "desc", for the lowest value to retrieve rank 1, use "asc".</p> <p>Ties are given rank values depending on optional argument values:</p> <p>"ties.method=minimum" (default),</p> <p>"ties.method=maximum",</p> <p>"ties.method=first", or</p> <p>"ties.method=average".</p> <p>See <i>More about ranking ties</i> below for more information about the available arguments. The average ties method is used when calculating data relationships using Spearman R.</p> <p>Additional column arguments (optional) can be used when the column should be split into separately ranked categories.</p> <p>Examples:</p> <p>RankReal([Sales])</p> <p>RankReal([Sales], "desc", [Region])</p> <p>RankReal([Sales], "ties.method=average")</p>



If an axis using a custom expression with ranking has been set up to Evaluate axis expression on: Current filtering only, then the ranking will be recalculated with each filtering.

### More about ranking ties:

With the functions Rank and RankReal, you can add an optional ties method argument depending how you want equal values to be ranked.

Argument	Description
"ties.method=minimum"	Gives all ties the smallest rank value of the tie values.
"ties.method=maximum"	Gives all ties the largest rank value of the tie values.
"ties.method=first"	Gives the first found tie value the lowest rank value, and continues with the following rank value for the next tie.
"ties.method=average"	Gives all ties the average of the rank values for all ties.

### Example:

When a list is to be ranked, its values are first of all sorted. Then, the sorted values are assigned a rank value depending on the order in the sorted list. What rank is given to a tie value depends on the ties method. Empty values are left empty and do not receive any rank.

List of values	Rank with "ties.method=minimum"	Rank with "ties.method=maximum"	Rank with "ties.method=first"	Rank with "ties.method=average"
1	1	1	1	1

2	2	3	2	2.5
3	4	4	4	4
2	2	3	3	2.5
(Empty)	(Empty)	(Empty)	(Empty)	(Empty)
5	5	5	5	5

If DenseRank was used, the resulting rank values in the example would be 1 2 3 4.

See also [Functions](#).

## Spatial functions

Use the spatial functions to transform data so it can be used to create map charts.

If the map information is included in a shapefile or GeoJSON file, you do not need to use the spatial functions. However, if you have geographic information in some other type of BLOB column containing WKB (Well-Known Binary) data, then this information needs to be extracted into seven different columns

- Geometry
- XMax
- XMin
- YMax
- YMin
- XCenter
- YCenter



The Geometry column is the original, binary column.

The bounding box for a geometry is called the envelope. It is specified by the four coordinates XMax, XMin, YMax and YMin. The center of the geometry is specified by the two coordinates XCenter and YCenter. These coordinate columns can be calculated from the binary WKB column using the spatial functions with the binary WKB column as an argument. In order for the map chart to identify these columns, they must also have the required property values (the same as the column names listed above)

set on the `mapchart.columntypeid` property. This is automatically done when the spatial functions below are applied.

Function	Description
<code>GreatCircleDistance(Arg1, Arg2, Arg3, Arg4)</code>	<p>Returns the shortest distance between two points, calculated on the surface of a unit sphere. The arguments are the latitude and longitude of the coordinates (lat1, long1, lat2, long2), expressed in degrees.</p> <p>To obtain the result in a certain unit, multiply the result from the <code>GreatCircleDistance</code> method with the radius of the interesting sphere in the desired unit. For example, the radius of the earth is approximately 6371 km (3959 miles), so the first example below will give the distance between two cities expressed in kilometers and the second example gives it in miles.</p> <p>Examples:</p> <pre>6371*GreatCircleDistance(57.717829, 11.990509, 42.355145, -71.057892) 3959*GreatCircleDistance(57.717829, 11.990509, 42.355145, -71.057892) 6371*GreatCircleDistance([Latitude], [Longitude], 42.355145, -71.057892)</pre>
<code>WKBEnvelopeXCenter(Arg1)</code>	<p>Calculates the X center of the geometry envelope and sets the XCenter value on the <code>mapchart.columntypeid</code> property. The argument is a binary WKB column.</p> <p>Example:</p> <pre>WKBEnvelopeXCenter([WKB])</pre>
<code>WKBEnvelopeXMin(Arg1)</code>	<p>Calculates the X min of the geometry envelope and sets the XMin value on the <code>mapchart.columntypeid</code> property. The argument is a binary WKB column.</p> <p>Example:</p> <pre>WKBEnvelopeXMin([WKB])</pre>
<code>WKBEnvelopeXMax(Arg1)</code>	<p>Calculates the X max of the geometry envelope and sets the XMax value on the <code>mapchart.columntypeid</code> property. The argument is a binary WKB column.</p> <p>Example:</p> <pre>WKBEnvelopeXMax([WKB])</pre>
<code>WKBEnvelopeYCenter(Arg1)</code>	<p>Calculates the Y center of the geometry envelope and sets the YCenter value on the <code>mapchart.columntypeid</code> property. The argument is a binary WKB column.</p> <p>Example:</p> <pre>WKBEnvelopeYCenter([WKB])</pre>
<code>WKBEnvelopeYMin(Arg1)</code>	<p>Calculates the Y min of the geometry envelope and sets the YMin value on the <code>mapchart.columntypeid</code> property. The argument is a binary WKB column.</p> <p>Example:</p> <pre>WKBEnvelopeYMin([WKB])</pre>


Function	Description
<code>WKBEvelopeYMax(Arg1)</code>	<p>Calculates the Y max of the geometry envelope and sets the YMax value on the <code>mapchart.columntypeid</code> property. The argument is a binary WKB column.</p> <p>Example:</p> <pre>WKBEvelopeYMax([WKB])</pre>

See also [Functions](#).

## Statistical functions

The list shows the statistical functions that you can use in expressions.

Function	Description
<code>Avg(Arg1, ...)</code>	<p>Returns the average (arithmetic mean) of the arguments. The arguments and the result are of type real. If one argument is given, then the result is the average of all rows. If more than one argument is given, then the result is the average for each row. Null arguments are ignored and do not contribute to the average.</p> <p>Examples:</p> <pre>Avg([Column]) Avg(2, -3, 4) → 1 Avg(-1) → -1 Avg(1.5, -2, 3.5) → 1 Avg(1, null, 3) → 2 Avg(null) → (Empty)</pre>
<code>ChiDist(Arg1)</code>	<p>Returns the (upper tail) chi-square p-value of the argument.</p> <p>Example:</p> <pre>ChiDist(x, deg_freedom) ChiDist(7.377759, 2) → 0.025</pre>
<code>ChiInv(Arg1)</code>	<p>Returns the (upper tail) chi-square quantile value of the argument.</p> <p>Example:</p> <pre>ChiInv(p, deg_freedom) ChiInv(0.025, 2) → 7.377759</pre>
<code>Count(Arg1)</code>	<p>Calculates the number of non-empty values in the argument column, or, if no argument is specified, the total number of rows.</p> <p>Example:</p> <pre>Count([Column])</pre>
<code>CountBig(Arg1)</code>	<p>Calculates the number of non-empty values in the argument column, or, if no argument is specified, the total number of rows. This function returns a LongInteger.</p> <p>Example:</p> <pre>CountBig([Column])</pre>

Function	Description
<code>Covariance(Arg1, Arg2)</code>	<p>Calculates the covariance of two columns given as arguments.</p> <p>Example:</p> <pre>Covariance([Column1], [Column2])</pre>
<code>FDist(Arg1)</code>	<p>Returns the upper tail F p-value of the argument.</p> <p>Example:</p> <pre>FDist(x, deg_freedom1, deg_freedom2)</pre> <pre>FDist(6.936728, 1, 10) → 0.025</pre>
<code>FInv(Arg1)</code>	<p>Returns the upper tail F quantile value of the argument.</p> <p>Example:</p> <pre>FInv(p, deg_freedom1, deg_freedom2)</pre> <pre>FInv(0.025, 1, 10) → 6.936728</pre>
<code>First(Arg1)</code>	<p>Returns the first valid value based on the physical order of the rows of data in the argument column.</p> <p>Example:</p> <pre>First([Column])</pre>
<code>GeometricMean()</code>	<p>Calculates the geometric mean value. If any input value is negative then the result will be "Empty". If any input value is equal to zero then the result will be zero.</p> <p>Example:</p> <pre>GeometricMean([Sales])</pre>
<code>IQR(Arg1)</code>	<p>Calculates the value difference Q3-Q1, or, the 75th percentile minus the 25th percentile. IQR is also referred to as the H-spread.</p> <p>Example:</p> <pre>IQR([Column])</pre>
<code>L95(Arg1)</code>	<p>Calculates the lower endpoint of the 95% confidence interval.</p> <div>  <p>The static t-value of 1.959964, used by this function, is adapted for large sample sizes (<math>n \geq 40</math>). For smaller sample sizes, use the following expression instead:</p> <pre>Avg([Value]) - TInv(0.025, Count() - 1) * StdDev([Value]) / Sqrt(Count())</pre> </div> <p>Example:</p> <pre>L95([Column])</pre>




Function	Description																
<code>Lag(Arg1, Arg2)</code>	<p>Shifts the values in a column downward by a specified number of steps. The first argument is the column to shift. The second (optional) argument is the number of steps. Default is 1.</p> <p>If a negative number of steps is used, then the values are shifted in the opposite direction, see image below.</p> <p>Examples:</p> <pre>Lag([Column])</pre> <pre>Lag([Column], 3)</pre> <table><thead><tr><th>ID</th><th>Lag([ID])</th><th>Lag([ID],2)</th><th>Lag([ID], -1)</th></tr></thead><tbody><tr><td>1</td><td></td><td></td><td>2</td></tr><tr><td>2</td><td>1</td><td></td><td>3</td></tr><tr><td>3</td><td>2</td><td>1</td><td></td></tr></tbody></table> <p>Note that the Lag function is applied to the data in the order the data was loaded; the function does not take sorting in visualizations into account and any changes to the data (e.g., during reload) might result in different values for the various rows.</p>	ID	Lag([ID])	Lag([ID],2)	Lag([ID], -1)	1			2	2	1		3	3	2	1	
ID	Lag([ID])	Lag([ID],2)	Lag([ID], -1)														
1			2														
2	1		3														
3	2	1															
<code>Last(Arg1)</code>	<p>Returns the last valid value based on the physical order of the rows of data in the argument column.</p> <p>Example:</p> <pre>Last([Column])</pre>																
<code>LastValueForMax(Arg1, Arg2)</code>	<p>Returns the value of column 2 for the maximum value of column 1.</p> <p>If there are more than one of the column 1 maximum value, then the result will be the value for the last max row. See also <code>ValueForMax</code>.</p> <p>Example:</p> <pre>LastValueForMax([Column 1], [Column 2])</pre>																
<code>LastValueForMin(Arg1, Arg2)</code>	<p>Returns the value of column 2 for the minimum value of column 1.</p> <p>If there are more than one of the column 1 minimum value, then the result will be the value for the last min row. See also <code>ValueForMin</code>.</p> <p>Example:</p> <pre>LastValueForMin([Column 1], [Column 2])</pre>																
<code>LAV(Arg1)</code>	<p>Calculates the lower adjacent value.</p> <p>Example:</p> <pre>LAV([Column])</pre>																

Function	Description																
<code>Lead(Arg1, Arg2)</code>	<p>Shifts the values in a column upward by a specified number of steps. The first argument is the column to shift. The second (optional) argument is the number of steps. Default is 1.</p> <p>If a negative number of steps is used, then the values are shifted in the opposite direction, see image below.</p> <p>Examples:</p> <pre>Lead([Column])</pre> <pre>Lead([Column], 3)</pre> <table><thead><tr><th>ID</th><th>Lead([ID], 1)</th><th>Lead([ID], 2)</th><th>Lead([ID], -1)</th></tr></thead><tbody><tr><td>1</td><td>2</td><td>3</td><td></td></tr><tr><td>2</td><td>3</td><td></td><td>1</td></tr><tr><td>3</td><td></td><td></td><td>2</td></tr></tbody></table> <p>Note that the Lead function is applied to the data in the order the data was loaded; the function does not take sorting in visualizations into account and any changes to the data (e.g., during reload) might result in different values for the various rows.</p>	ID	Lead([ID], 1)	Lead([ID], 2)	Lead([ID], -1)	1	2	3		2	3		1	3			2
ID	Lead([ID], 1)	Lead([ID], 2)	Lead([ID], -1)														
1	2	3															
2	3		1														
3			2														
<code>LIF(Arg1)</code>	<p>Calculates the lower inner fence. This is the threshold located at <math>Q1 - (1.5 * IQR)</math>.</p> <p>Example:</p> <pre>LIF([Column])</pre>																
<code>LOF(Arg1)</code>	<p>Calculates the lower outer fence. This is the threshold located at <math>Q1 - (3 * IQR)</math>.</p> <p>Example:</p> <pre>LOF([Column])</pre>																
<code>Max(Arg1, ...)</code>	<p>Calculates the maximum value. If one argument is given, then the result is the maximum for the entire column. If more than one argument is given, then the result is the maximum for each row. The argument and the result are of type real. Null arguments are ignored.</p> <p>Examples:</p> <pre>Max([Column])</pre> <pre>Max(-1) → -1</pre> <pre>Max(1.5, -2, 3) → 3</pre> <pre>Max(1, null, 3) → 3</pre> <pre>Max(null) → (Empty)</pre>																
<code>MeanDeviation(Arg1, ...)</code>	<p>Calculates the mean deviation value (average absolute deviation, AAD). If one argument is given, then the result is the mean deviation of all rows. If more than one argument is given, then the result is the mean deviation for each row.</p> <p>Examples:</p> <pre>MeanDeviation([Column])</pre> <pre>MeanDeviation(2, -3, 4) → 2.67</pre>																

Function	Description
<code>Median(Arg1)</code>	<p>Calculates the median of the argument. If one argument is given, then the result is the median of all rows. If more than one argument is given, then the result is the median for each row.</p> <p>Examples:</p> <pre>Median([Column]) Median(2,-3,4)</pre>
<code>MedianAbsoluteDeviation(Arg1, ...)</code>	<p>Calculates the median absolute deviation (MAD). If one argument is given, then the result is the median absolute deviation of all rows. If more than one argument is given, then the result is the median absolute deviation for each row.</p> <p>Examples:</p> <pre>MedianAbsoluteDeviation([Sales]) MedianAbsoluteDeviation(2,-3,4)</pre>
<code>Min(Arg1, ...)</code>	<p>Calculates the minimum value. If one argument is given, then the result is the minimum for the entire column. If more than one argument is given, then the result is the minimum for each row. The argument and the result are of type real. Null arguments are ignored.</p> <p>Examples:</p> <pre>Min([Column]) Min(-1) → -1 Min(1.5, -2, 3) → -2 Min(1, null, 3) → 1 Min(null) → (Empty)</pre>
<code>NormDist(Arg1)</code>	<p>Returns the (upper tail) normal p-value of the argument. If you do not specify them yourself, the default is mean=0 and standard deviation=1.</p> <p>Example:</p> <pre>NormDist(x, mean, standard_dev) NormDist(1.96) → 0.025</pre>
<code>NormInv(Arg1)</code>	<p>Returns the (upper tail) normal quantile value of the argument. If you do not specify them yourself, the default is mean=0 and standard deviation=1.</p> <p>Example:</p> <pre>NormInv(p, mean, standard_dev) NormInv(0.025) → 1.96</pre>
<code>NthLargest(Arg1, Arg2)</code>	<p>The nth largest value. The first argument is the column to analyze and the second argument is the value of n.</p> <p>If n is larger than the number of values in the column, then the smallest value is returned.</p> <p>Example:</p> <pre>NthLargest([Column], 10)</pre>

Function	Description
<code>NthSmallest(Arg1, Arg2)</code>	<p>The nth smallest value. The first argument is the column to analyze and the second argument is the value of n.</p> <p>If n is larger than the number of values in the column, then the largest value is returned.</p> <p>Example:</p> <pre>NthSmallest([Column], 10)</pre>
<code>Outliers(Arg1)</code>	<p>Outer value count. Calculates the count of values that are greater than the upper adjacent value or lower than the lower adjacent value.</p> <p>Example:</p> <pre>Outliers([Column])</pre>
<code>P10(Arg1)</code>	<p>The 10th percentile is the value at which 10 percent of the data values are equal to or lower than the value.</p> <p>Example:</p> <pre>P10([Column])</pre>
<code>P90(Arg1)</code>	<p>The 90th percentile is the value at which 90 percent of the data values are equal to or lower than the value.</p> <p>Example:</p> <pre>P90([Column])</pre>
<code>PctOutliers(Arg1)</code>	<p>Outer value percentile. Calculates the percent of values that are greater than the upper adjacent value or lower than the lower adjacent value.</p> <p>Example:</p> <pre>PctOutliers([Column])</pre>
<code>Percent(Arg1, Arg2)</code>	<p>The percent is the value calculated a certain percent above the minimum value within the value range (max value - min value). The first argument is the column to analyze and the second argument is the percent.</p> <p>Example:</p> <pre>Percent([Column], 15.0)</pre>
<code>Percentile(Arg1, Arg2)</code>	<p>The percentiles is the value at which a certain percent of the data values are equal to or lower than the value. The first argument is the column to analyze and the second argument is the percent.</p> <p>Example:</p> <pre>Percentile([Column], 15.0)</pre>
<code>Q1(Arg1)</code>	<p>Calculates the first quartile.</p> <p>Example:</p> <pre>Q1([Column])</pre>
<code>Q3(Arg1)</code>	<p>Calculates the third quartile.</p> <p>Example:</p> <pre>Q3([Column])</pre>

Function	Description
<code>Range(Arg1)</code>	<p>The range between the largest and the smallest value in the column.</p> <p>The result will be presented as a real or a timespan, depending on the data type of the argument.</p> <p>Example:</p> <pre>Range([Column])</pre>
<code>StdDev(Arg1)</code>	<p>Calculates the standard deviation.</p> <p>Example:</p> <pre>StdDev([Column])</pre>
<code>StdErr(Arg1)</code>	<p>Calculates the standard error.</p> <p>Example:</p> <pre>StdErr([Column])</pre>
<code>TDist(Arg1)</code>	<p>Returns the (upper tail) t p-value of the argument.</p> <p>Example:</p> <pre>TDist(x, deg_freedom)</pre> <p><code>TDist(4.302653, 2) → 0.025</code></p>
<code>TERR_Binary</code>	<p>Calls the Spotfire® Enterprise Runtime for R (a/k/a TERR™) and returns an output of the specified data type, containing the same number of rows as the input.</p> <p>The first argument is a script and the following arguments are the arguments to the script.</p> <p>The column returned must have the same number of rows as the input. At least one argument other than the script is required. The inputs will be placed in variables called <code>input1</code>, <code>input2</code>, ... <code>inputN</code>, and the output must be placed in a variable called <code>output</code>.</p> <p>Examples:</p> <pre>TERR_Real("output &lt;- input1*100 + input2", [Record No], [Sales])</pre> <pre>TERR_String("output &lt;- input1", [String Column])</pre>
<code>TERR_Boolean</code>	See <code>TERR_Binary</code> above.
<code>TERR_DateTime</code>	See <code>TERR_Binary</code> above.
<code>TERR_Integer</code>	See <code>TERR_Binary</code> above.
<code>TERR_Real</code>	See <code>TERR_Binary</code> above.
<code>TERR_String</code>	See <code>TERR_Binary</code> above.

Function	Description
TERRAggregation_Binary	<p>Calls the TERR engine and returns an output of the specified data type. The first argument is a script and the following arguments are the arguments to the script.</p> <p>The script should return a single aggregated value. At least one argument other than the script is required. The inputs will be placed in variables called input1, input2, ... inputN, and the output must be placed in a variable called output.</p> <p>Examples:</p> <pre>TERRAggregation_Real("output &lt;- median(input1) + median(input2)", [X], [Y])</pre> <pre>TERRAggregation_String("output &lt;- input1[1]", [Customer Name])</pre>
TERRAggregation_Boolean	See TERRAggregation_Binary above.
TERRAggregation_DateTime	See TERRAggregation_Binary above.
TERRAggregation_Integer	See TERRAggregation_Binary above.
TERRAggregation_Real	See TERRAggregation_Binary above.
TERRAggregation_String	See TERRAggregation_Binary above.
TInv(Arg1)	<p>Returns the (upper tail) t quantile value of the argument.</p> <p>Examples:</p> <pre>TInv(p, deg_freedom)</pre> <pre>TInv(0.025, 2) → 4.302653</pre>
TrimmedMean(Arg1, Arg2)	<p>Calculates the trimmed mean value (trimmed average). The first argument is the column to analyze and the second argument is, in percent, how many values to exclude from the calculation. If the trim value is set to 10%, then the highest 5% and the lowest 5% of the values are excluded from the calculated mean.</p> <p>Example:</p> <pre>TrimmedMean([Sales], 10)</pre>
U95(Arg1)	<p>Calculates the upper endpoint of the 95% confidence interval.</p> <div>  <p>The static t-value of 1.959964, used by this function, is adapted for large sample sizes (<math>n \geq 40</math>). For smaller sample sizes, use the following expression instead:</p> <pre>Avg([Value]) + TInv(0.025, Count() - 1) * StdDev([Value]) / Sqrt(Count())</pre> </div> <p>Example:</p> <pre>U95([Column])</pre>
UAV(Arg1)	<p>Calculates the upper adjacent value.</p> <p>Example:</p> <pre>UAV([Column])</pre>

Function	Description
<code>UIF(Arg1)</code>	<p>Calculates the upper, inner fence. This is the threshold located at <math>Q3 + (1.5 * IQR)</math>.</p> <p>Example:</p> <p><code>UIF([Column])</code></p>
<code>UniqueCount(Arg1)</code>	<p>Calculates the number of unique, non-empty values in the argument column.</p> <p>Example:</p> <p><code>UniqueCount([Column])</code></p>
<code>UOF(Arg1)</code>	<p>Calculates the upper outer fence. This is the threshold located at <math>Q3 + (3 * IQR)</math>.</p> <p>Example:</p> <p><code>UOF([Column])</code></p>
<code>ValueForMax(Arg1, Arg2)</code>	<p>Returns the value of column 2 for the maximum value of column 1.</p> <p>If there are more than one of the column 1 maximum value, then the result will be the value for the first max row. See also <code>LastValueForMax</code>.</p> <p>Example:</p> <p><code>ValueForMax([Column 1], [Column 2])</code></p>
<code>ValueForMin(Arg1, Arg2)</code>	<p>Returns the value of column 2 for the minimum value of column 1.</p> <p>If there are more than one of the column 1 minimum value, then the result will be the value for the first min row. See also <code>LastValueForMin</code>.</p> <p>Example:</p> <p><code>ValueForMin([Column 1], [Column 2])</code></p>
<code>Var(Arg1)</code>	<p>Calculates the variance.</p> <p>Example:</p> <p><code>Var([Column])</code></p>
<code>WeightedAverage(Arg1, Arg2)</code>	<p>Calculates the weighted average of two columns. Arg1 is the weight column and Arg2 is the value column.</p> <p>Example:</p> <p><code>WeightedAverage([Column1],[Column2])</code></p>

**Tip:** The `DISTINCT` keyword can be used to return a result using the unique values only. For example, `Avg(DISTINCT[Column])` would return the average of the unique values rather than the average of all values in the specified column. `UniqueCount([Column])` is the equivalent of `Count(DISTINCT[Column])`.

See also [Functions](#).

## Text functions

The list shows the text functions that you can use in expressions.

Function	Description
~=	<p>Can be part of an If or Case statement. Returns true if the <i>Arg2</i> regular expression string matches the <i>Arg1</i> string.</p> <p>Examples:</p> <pre>If( "aab" ~= "a+" , "true", "false" )</pre> <pre>Case when "aba" ~= ".a+\$" then "true" else "false" end</pre>
Concatenate( <i>Arg1</i> , ...)	<p>Concatenates (appends) all the arguments into a string. If one argument is given, then the result is the concatenation of all rows. If more than one argument is given, then each row is concatenated. The arguments can be of any type, but are converted to strings. The result is of type string. Null arguments are ignored.</p> <p>Examples:</p> <pre>Concatenate("April ", 20+1, "st") → "April 21st"</pre> <pre>Concatenate(null, "Ape") → "Ape"</pre> <pre>Concatenate (null, null) → (Empty)</pre>
Find( <i>Arg1</i> , <i>Arg2</i> , <i>Arg3</i> )	<p>Returns the index of the occurrence of the string <i>Arg1</i> in <i>Arg2</i>. <i>Arg3</i> optionally indicates which match to return. If no third argument is given then the first match is returned. If no match is found, 0 is returned.</p> <p>The search is case-sensitive. The first two arguments are of type string and the third argument and the result are of type integer. If <i>Arg1</i> is the empty string, 0 is returned.</p> <p>Examples:</p> <pre>Find("lo", "Hello") → 4</pre> <pre>Find("a", "Hello") → 0</pre> <pre>Find("", "Hello") → 0</pre> <pre>Find("", null) → (Empty)</pre> <pre>If(Find("Pri 1", [Coll])&gt;0, "Important", "Not important")</pre> <pre>Find("a", "ababab") → 1</pre> <pre>Find("a", "ababab", 2) → 3</pre> <pre>Find("a", "ababab", 10) → 0</pre>
Left( <i>Arg1</i> , <i>Arg2</i> )	<p>Returns the first <i>Arg2</i> characters of the string <i>Arg1</i>. <i>Arg1</i> and the result are of type string. <i>Arg2</i> is of type real, but only the integer part is used. If <i>Arg2</i> &gt; the length of <i>Arg1</i>, the whole string is returned. If <i>Arg2</i> is negative, then the <i>Arg2</i> number of characters will be removed from the right-hand side of the <i>Arg1</i> string.</p> <p>Examples:</p> <pre>Left("Daddy", 1) → "D"</pre> <pre>Left("Daddy", 3.99) → "Dad"</pre> <pre>Left("Daddy", 386) → "Daddy"</pre> <pre>Left("Daddy", -1) → "Dadd"</pre>



Function	Description
<code>Len(Arg1)</code>	<p>Returns the length of <i>Arg1</i>. <i>Arg1</i> is of type string and the result is of type integer.</p> <p>Examples:</p> <p><code>Len("Hello") → 5</code></p> <p><code>Len(null) → (Empty)</code></p>
<code>Lower(Arg1)</code>	<p>Returns <i>Arg1</i> converted to lowercase. <i>Arg1</i> and the result are of type string.</p>
<code>Mid(Arg1, Arg2, Arg3)</code>	<p>Returns the substring of <i>Arg1</i> starting at index <i>Arg2</i> with a length of <i>Arg3</i> characters. <i>Arg1</i> and the result are of type string. <i>Arg2</i> and <i>Arg3</i> are of type real, but only the integer part is used. If <i>Arg2</i> &gt; <code>Len(Arg1)</code>, an empty string is returned. Else, if <i>Arg2</i>+<i>Arg3</i> &gt; <code>Len(Arg1)</code>, <i>Arg3</i> is adjusted to <code>1+Len(Arg1)-Arg2</code>. If either of <i>Arg2</i> or <i>Arg3</i> is negative or if <i>Arg2</i> is zero, an error is returned.</p> <p>Examples:</p> <p><code>Mid("Daddy", 2, 3) → "add"</code></p> <p><code>Mid("Daddy", 386, 4) → ""</code></p> <p><code>Mid("Daddy", 4, 386) → "dy"</code></p> <p><code>Mid("Daddy", -1, 2) → (Empty)</code></p> <p><code>Mid("Daddy", 2, -1) → (Empty)</code></p>
<code>MostCommon(Arg1)</code>	<p>Returns the most common value of the specified column. If several values are equally common, the first one will be used.</p> <p>Example:</p> <p><code>MostCommon([Column])</code></p>
<code>NameDecode(Arg1)</code>	<p>Replaces all substring codes with decoded characters.</p> <p>Column names in Spotfire are stored as UTF-16 encoded strings.</p> <p>Example:</p> <p><code>NameDecode("Column %02D")</code></p>
<code>NameEncode(Arg1)</code>	<p>Encodes characters in the string so that the string only contains characters matching the regular expression <code>[.0-9a-zA-Z]</code>.</p> <p>Column names in Spotfire are stored as UTF-16 encoded strings.</p> <p>Example:</p> <p><code>NameEncode("Column £")</code></p>
<code>Repeat</code>	<p>Repeats a string a specified number of times.</p> <p>Example:</p> <p><code>Repeat("Hello", 2) → "HelloHello"</code></p>

Function	Description
<code>Right(Arg1, Arg2)</code>	<p>Returns the last <i>Arg2</i> characters of the string <i>Arg1</i>. <i>Arg1</i> and the result are of type string. <i>Arg2</i> is of type real, but only the integer part is used. If <i>Arg2</i> &gt; the length of <i>Arg1</i>, the whole string is returned.</p> <p>If <i>Arg2</i> is negative, then the <i>Arg2</i> number of characters will be removed from the left-hand side of the <i>Arg1</i> string.</p> <p>Examples:</p> <pre>Right("Daddy", 1) → "y" Right("Daddy", 3.99) → "ddy" Right("Daddy", 386) → "Daddy" Right("Daddy", -1) → "addy"</pre>
<code>RXExtract(Arg1, Arg2, Arg3)</code>	<p>Returns the part of a string (<i>Arg1</i>) that matches a regular expression (<i>Arg2</i>). <i>Arg3</i> determines which match to use when there are multiple matches.</p> <p>Examples:</p> <pre>RXExtract([Column], "l+", 1) RXExtract("Parallel", "l+", 1) → "ll" RXExtract("Parallel", "l+", 2) → "l"</pre>
<code>RXReplace(Arg1, Arg2, Arg3, Arg4)</code>	<p>Replaces a substring according to a regular expression. Search for the <i>Arg2</i> regular expression in <i>Arg1</i> and replace it with <i>Arg3</i>.</p> <p><i>Arg4</i> specifies the options for the replacement:</p> <p>"g" specifies that if <i>Arg2</i> matches more than once then all matches should be substituted.</p> <p>"i" specifies that the comparison should be case insensitive.</p> <p>"s", for single-line mode, specifies that the dot (.) matches every character (instead of every character except newline).</p> <p>Some characters, like for instance the backslash character "\", and parentheses, "(" and ")", need to be escaped to work when using calculated columns. Because parenthesis are used for grouping in regular expressions, both "(" and ")" need to be escaped if the ( or ) sign should be matched, and when the character is used in a regular expression double backslashes are needed in order to escape both the Spotfire string and the regular expression. See literature about regular expression language elements, e.g., on <a href="#">MSDN</a>, for more information.</p> <p>Examples:</p> <pre>RXReplace("Hello", "L+", "LL", "i") → "HeLLo" RXReplace("3 Minor", "(\\d).*", "\$1", "") → 3 RXReplace([Column with values within parentheses], "\\(89\\)", "", "") RXReplace("change\\slashdirection", "\\\\", "/", "") → change/ slashdirection</pre> <p>(In the last example, the backslash needs to be escaped twice; once for the Spotfire string and once for the regular expression.)</p>

Function	Description
<code>Split(Arg1, Arg2, Arg3)</code>	<p>Splits the values in <i>Arg1</i> into a number of substrings, using <i>Arg2</i> as the separator. <i>Arg3</i> specifies which of the values should be returned as a new column. Only one column is returned.</p> <p>Use a negative <i>Arg3</i> to specify that the split should be done from the end of the string toward the beginning, instead of from the beginning toward the end.</p> <p>For example, if a column called "X" containing the value "a.b" is split into two parts using the separator ".", then <code>Split([X], ".", 1)</code> will return "a" and <code>Split([X], ".", 2)</code> will return "b".</p> <p>Examples:</p> <pre>Split([Column], " ", 2) Split([Column], " ", -2) Split([Column], " / ", 3)</pre>
<code>Substitute(Arg1, Arg2, Arg3)</code>	<p>Replaces all occurrences of <i>Arg2</i> in <i>Arg1</i> with <i>Arg3</i>. The search is case sensitive.</p> <p>Example:</p> <pre>Substitute("Test", "t", "ting") → "Testing"</pre>
<code>Substring(Arg1, Arg2, Arg3)</code>	<p>Returns the substring of <i>Arg1</i> starting at index <i>Arg2</i> with a length of <i>Arg3</i> characters. <i>Arg1</i> and the result are of type string. <i>Arg2</i> and <i>Arg3</i> are of type real, but only the integer part is used. If <math>Arg2 &gt; Len(Arg1)</math>, an empty string is returned. Else, if <math>Arg2 + Arg3 &gt; Len(Arg1)</math>, <i>Arg3</i> is adjusted to <math>1 + Len(Arg1) - Arg2</math>. If either of <i>Arg2</i> or <i>Arg3</i> is negative or if <i>Arg2</i> is zero, an error is returned.</p> <p>Examples:</p> <pre>Substring("Daddy", 2, 3) → "add" Substring("Daddy", 386, 4) → "" Substring("Daddy", 4, 386) → "dy" Substring("Daddy", -1, 2) → (Empty) Substring("Daddy", 2, -1) → (Empty)</pre>
<code>Trim(Arg1)</code>	<p>Removes whitespace characters from the beginning and end of a string.</p> <p>Example:</p> <pre>Trim(" Example ") → "Example"</pre>
<code>UniqueConcatenate(Arg1)</code>	<p>Concatenates the unique values converted to strings. The values are ordered according the comparator.</p> <p>Example:</p> <pre>UniqueConcatenate([Column])</pre>
<code>Upper(Arg1)</code>	<p>Returns <i>Arg1</i> converted to uppercase. <i>Arg1</i> and the result are of type string.</p> <p>Example:</p> <pre>Upper("hello") → "HELLO"</pre>

See also [Functions](#).

## Data types

The list shows the available data types.

Data type	Description
<b>Integer</b>	<p>Integer values are written as a sequence of digits, possibly prefixed by a + or - sign. The integer values that can be specified range from -2147483648 to 2147483647. If used where a decimal value was expected, the integer values are automatically converted to decimal values.</p> <p>Note that hexadecimal values can be used in custom expressions and in calculated columns. They cannot be used when opening data. Hexadecimal-formatted values have a size limitation of 8 characters.</p> <p><b>Examples:</b></p> <p>0</p> <p>101</p> <p>-32768</p> <p>+55</p> <p>0xff = 255</p> <p>0x7fffffff = 2147483647</p> <p>0x80000000 = -2147483648</p>
<b>LongInteger</b>	<p>LongInteger can be used if the range for the standard Integer is not enough for your needs. It ranges from -9223372036854775808 to 9223372036854775807. LongInteger cannot be converted to Real without precision loss, but it can be converted to Currency without precision loss.</p> <p>Note: Hexadecimal values can be used in custom expressions and in calculated columns. They cannot be used when opening data.</p> <p><b>Examples:</b></p> <p>2147483648</p> <p>0x7FFFFFFFFFFFFFFF = 9223372036854775807</p> <p>0x8000000000000000 = -9223372036854775808</p>
<b>Real</b>	<p>Real values are written as standard floating point numbers with a period for a decimal point and no thousands separator. The real values that can be specified range from -8.98846567431157E+307 to 8.98846567431157E+307.</p> <p>The number of significant digits that can be shown is limited to 15, even though 16 can be used in calculations.</p> <p>Math operations on real values which produce results that cannot be represented by the real data type generate numeric errors. In the resulting data table, these special cases will be filtered out and replaced by null.</p> <p><b>Examples:</b></p> <p>0.0</p> <p>0.1</p> <p>10000.0</p> <p>-1.23e-22</p> <p>+1.23e+22</p> <p>1E6</p>

Data type	Description
<b>SingleReal</b>	<p>SingleReal values are written as standard floating point numbers with lower precision and range than Real. SingleReal occupies 50% less memory than Real. The SingleReal values that can be specified range from -1.7014117E+38 to 1.7014117E+38.</p> <p>The number of significant digits that can be shown is limited to 7, even though 8 can be used in calculations.</p> <p>SingleReal can be converted to Real with minor precision loss.</p>
<b>Currency</b>	<p>Currency constants are written as integer or real constants with an 'm' suffix.</p> <p>The data format behind the currency type is decimal. The decimal data format uses the base 10 in its calculations, which means that the round-off errors that may occur when doing binary calculations can be avoided with this format. However, this also means that heavy calculations take a longer time.</p> <p>The number of significant digits that can be shown for a currency value is 28 (29 can be used in calculations). Currency values that can be specified range from -39614081257132168796771975168 to 39614081257132168796771975168.</p> <p>Currency columns cannot be used in data functions.</p>
<b>Date</b>	<p>A date and time format depending on the locale on your computer. Dates from January 1, 1583 and forward are supported.</p> <p><b>Examples:</b></p> <p>6/12/2006</p> <p>June 12</p> <p>June, 2006</p>
<b>DateTime</b>	<p>A date and time format depending on the locale on your computer. Dates from January 1, 1583 and forward are supported.</p> <p><b>Examples:</b></p> <p>6/12/2006</p> <p>Monday, June 12, 2006 1:05 PM</p> <p>6/12/2006 10:14:35 AM</p>
<b>Time</b>	<p>A date and time format depending on the locale on your computer.</p> <p><b>Examples:</b></p> <p>2006-06-12 10:14:35</p> <p>10:14</p> <p>10:14:35</p>

Data type	Description
<b>TimeSpan</b>	<p>TimeSpan is a value describing the difference between two dates.</p> <p>It has 5 possible fields:</p> <p>Days</p> <ol style="list-style-type: none"> <li>Min: -10675199</li> <li>Max: 10675199</li> </ol> <p>Hours</p> <ol style="list-style-type: none"> <li>Min: 0</li> <li>Max: 23</li> </ol> <p>Minutes</p> <ol style="list-style-type: none"> <li>Min: 0</li> <li>Max: 59</li> </ol> <p>Seconds</p> <ol style="list-style-type: none"> <li>Min: 0</li> <li>Max: 59</li> </ol> <p>Fractions (decimals of seconds)</p> <ol style="list-style-type: none"> <li>Up to three decimals, i.e., the precision is 1 ms.</li> </ol> <p>TimeSpan values can be displayed on a compact form: [-]d.h:m:s.f ([-]days.hours:minutes.seconds.fractions) or written out with words or abbreviations for each available field. Some of the descriptive forms can be localized.</p> <p>Total min: -10675199.02:48:05.477</p> <p>Total max: 10675199.02:48:05.477</p>
<b>Boolean</b>	<p>True and false. Booleans are used to represent true and false values returned by comparison operators and logical functions.</p> <p>The display values can be localized.</p> <p><b>Examples:</b></p> <p>true</p> <p>false</p> <p>1 &lt; 5</p>
<b>String</b>	<p>String values are surrounded by double quotes or single quotes. Escaping is performed by entering the delimiter symbol twice in a row (i.e., " or ''"). A string value can contain any sequence of UNICODE characters. A double quote cannot be used within the string unless it is escaped. Backslash is used to escape special characters, so it too must be escaped.</p> <p>The basic escaping rules are that only the characters defined below can be used after a \; everything else will generate an error.</p> <p><b>Examples:</b></p> <p>"Hello world"</p> <p>"25""23"</p> <p>"1\n2\n"</p> <p>"C:\\TEMP\\image.png"</p>

Data type	Description
Binary	May contain any type of data, encoded in binary form.  <b>Examples:</b> Images Chemical structure information

## Properties in expressions

There are three types of properties available in Spotfire that you can use in expressions and to control different settings: document properties, data table properties and column properties. The properties can be interpreted in two different ways, either as actual values or as text values that can be used instead of column names or measures.

For general information about the different types of properties, see [Document properties, column properties and data table properties](#) on page 243.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client. You can write all of the expressions mentioned in the web client, but the installed client has some shortcuts and interfaces that are not available in the browser.

### Insert as Text

When you want to use a string property in an expression where it should be interpreted as a column name or a measure you must use a specific syntax. This is sometimes called a preprocessor expression.



This functionality was introduced to be able to easily add document property controls in text areas (installed client only), and to allow dashboard users to use these controls to change columns, etc., and update visualizations in the analysis. If you just want to use the value of a property in an expression, see **Insert as Value** below instead. If the expression can be written using the standard property functions, for example, `DocumentProperty()`, instead of using a preprocessor expression, then this is generally preferred.

When inserting as text, the name of the property should be enclosed in the following: "\${" and "}". So, for a document property this could look something like: `${MyProperty}`. For a data table property, the first part should be the data table name within the symbols specified above, followed by a period and the property name also within curly brackets: `${My Data Table}.MyProperty`. For a column property, the column name is also required: `${My Data Table}.Column Name.MyProperty`.



If any changes to an item used by the automatically added preprocessor syntax are required, you must manually update the expression. For example, if the name of a column or data table used in a preprocessor expression is changed (e.g., if you have the expression `[MyColumn] * ${Multiplier}` and the name of MyColumn is changed), you must manually update the expression to use the new name. This is because expressions using the preprocessor (\$-syntax) cannot be automatically deconstructed into individual parts, such as column names or data table names, the way properties used as values can. The preprocessor handles text before any parsing or type checking is performed and, as a result, it is not possible to know which parts of such a string are column names and which are not. The preprocessor can, however, be used to write expressions that are otherwise hard or impossible to write.

Property call in expression	Description
<code>\${MyProperty}</code>	Denotes the document property named MyProperty.

Property call in expression	Description
<code>\${My Data Table}. {MyProperty}</code>	Denotes the data table property named MyProperty defined for the data table "My Data Table".
<code>\${My Data Table} . {Column Name}. {MyProperty}</code>	Denotes the column property named MyProperty defined for the column "Column Name" in the data table "My Data Table".
<code>\$map("template", "concatenation string")</code>	<p>The <code>\$map()</code> function is used to map list-valued properties to a single string. The first argument is a template to use for each value in the list and the second argument is a specification of how the list values should be connected in the resulting expression.</p> <p>See <i>List-valued properties (multiple select)</i> below for more information.</p> <p>Examples:</p> <pre>\$map("sum([\${PropertyName}])", ",") &lt;\$map("\${PropertyName}", "NEST")&gt;</pre>
<code>\$esc(expression)</code>	<p>Replaces "]" in column names with "]]" and encloses the <a href="#">escaped</a> column names in "[" and "]". The argument is an expression which could be a property value or a property function that starts with a dollar sign (\$).</p> <p>Examples:</p> <pre>\$esc(\${MyProperty}) \$esc(\$csearch([Data Table], "Col*"))</pre>
<code>\$csearch([Data Table], "search string")</code>	<p>The <code>\$csearch()</code> function is used to select a number of columns from a data table using a limiting search expression. The first argument is a data table and the second argument is a string that contains the search expression determining which column names should be returned. The function returns a list of the (unescaped) column names from the data table that match the search expression.</p> <p>Examples:</p> <pre>\$csearch([Data Table], "**") \$csearch([Data Table], "Col*")</pre>

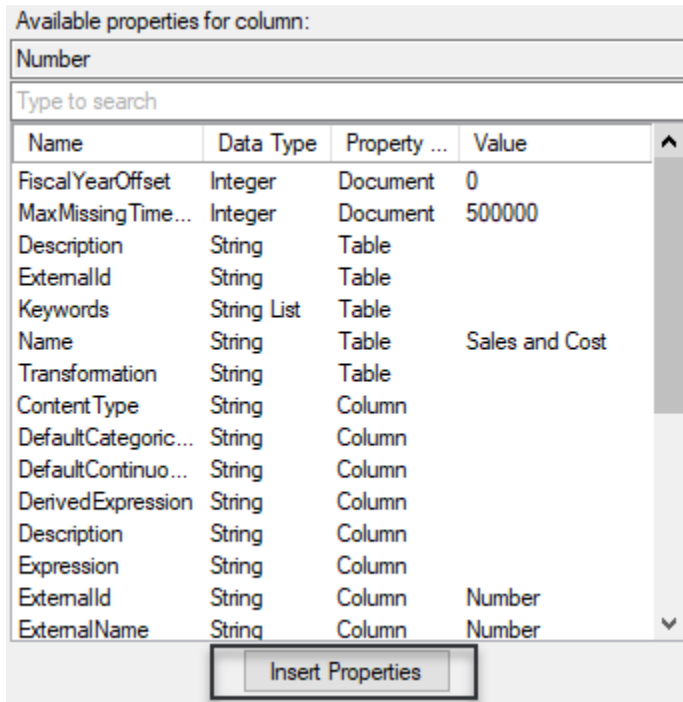
When the \$ sign is encountered in an expression, the property will be replaced by its corresponding value before the actual expression is evaluated (a preprocessing step). This means that if you type an expression like: `${Property1}-${Property2}`, and the value of Property1 is the column name Column1, and the value of Property2 is Column2, then the resulting expression will be: `[Column1]-[Column2]`.

If you do not want the property value to be interpreted as a column name but as a literal string value, you should put the property call within quotation marks. For example, if you have a property called "MyProperty" with the value "MyValue", the expression `${MyProperty}` would return the resulting expression MyValue, where a column called MyValue is retrieved. However, the expression `"${MyProperty}"` would return the text string "MyValue" instead. If the string value consists of several words (e.g., "My Value"), then you should use `$esc()` or put brackets around the property expression to return a column: `[${MyProperty}]`. See [Expression language details](#) on page 846 for more information about column names.

Note that it is very important to use the correct syntax for the expression at all times. Otherwise you might end up with a different result than you expected. If a property is inserted as a column where the expression expects a string, the first value of the specified column will be retrieved. For string columns, the result might be a valid expression, but for a column of another data type, you can instead receive an error.



In the installed client, if a property is inserted using the **Insert Properties** button in the **Add calculated column** dialog or in the Custom expression dialog, the property will automatically be added using the text (preprocessor) syntax.



You can also right-click the property in the **Available properties for column field** and select **Insert as Text** from the pop-up menu.

If a property containing a column name is to be used on an axis, there is a shortcut called **Set from Property** available from the pop-up menu on the column selectors (installed client only). If any changes to the automatically added syntax are required, you must use the Custom expression dialog to modify the expression. Right-click the column selector and select **Remove all** if you want to remove the property expression from the axis.

A property can also be used to specify an aggregation measure. For example, you can define a property called MyMeasure with the default value "sum". If such a property is to be used in an expression you must manually add parentheses and arguments for the measure in the expression where it is used. You can then create a property selector in a text area displaying a list of fixed aggregation measures that you want to be available in an axis expression and let web client users change the axis expression using the property selector.

Because string properties inserted as text in most cases will be interpreted as column names, you might encounter situations where you do not achieve the expected result when creating a custom expression. If the string property is to be interpreted as a value (for example, in conditional expressions) it must be surrounded by quotation marks. For example, if you want to replace the string "ProductA", used in an axis expression condition such as `Sum(if([Product])=("ProductA"),[Quantity],null))`, with a document property, then the document property must be surrounded by quotation marks to make the expression work:

```
Sum(if ([Product] = "${MyProduct}",[Quantity],null))
```

You can also insert the property as a function rather than as text to achieve the same results (see also *Insert as Value* below):

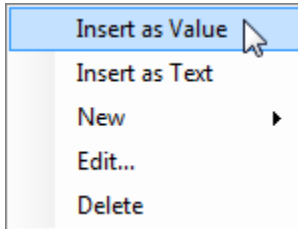
```
Sum(if ([Product] = DocumentProperty("MyProduct"),[Quantity],null))
```

## Insert as Value

When the value of a property is to be part of an expression, the recommendation is to use the standard [property functions](#): `ColumnProperty()`, `DataTableProperty()` and `DocumentProperty()` to encapsulate the property name. For example, use this syntax to write an expression with a document property holding an exchange rate times a value column expressed in some currency to be converted:

```
DocumentProperty("ExchangeRate")*[Value Column]
```

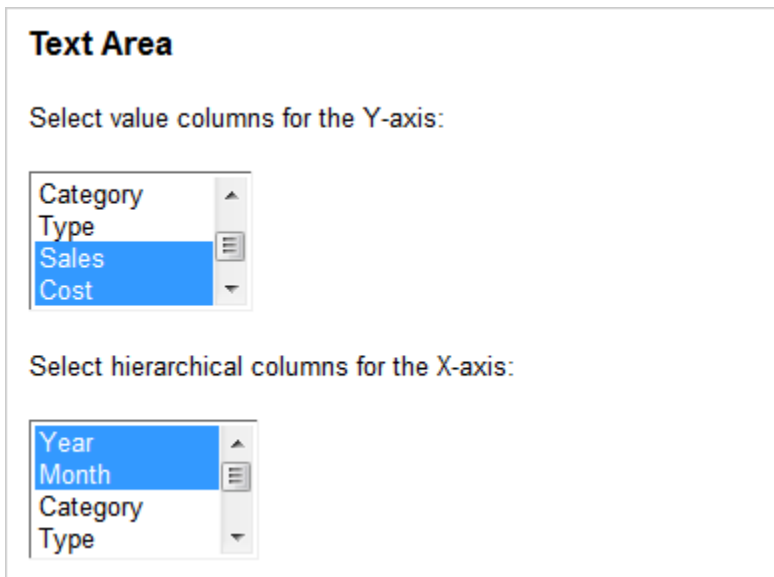
Note that the property name should always be written within quotation marks when property functions are applied. If you want a property to be interpreted as a value when using the installed client, you can right-click on the property in the **Available properties for column** field and select **Insert as Value** from the pop-up menu.



You can of course also manually type or edit the syntax for the property directly.

## List-valued properties (multiple select) – installed client only

Multiple-select properties, or list-valued properties, are based on a list of values instead of a single value. They can be created when defining a list box (multiple select) property control in a text area (installed client only). A list-valued property cannot be used in all places where a regular single-valued property can be applied, but it is an effective way to specify multiple columns on an axis.



List-valued properties often require some additional manual work when used in expressions. However, you will in most cases get a hint of what the expression should look like in the current context when inserting the property using the **Insert Properties** button.

When a list-valued property is added as text to a custom expression (for example, on a continuous axis like the Y-axis in a line chart, or the value axis in a bar chart) the syntax will be something like `$map("sum([{$PropertyNames}])", ", ")`. In this example, the default aggregation is set to "sum", so if you want to use a different aggregation you must manually edit the expression. The concatenation string is set to a comma, which means that each list value should

be interpreted as a separate column. If the `PropertyName` property contains a list with three columns: "ColumnA", "ColumnB" and "ColumnC", the resulting expression after expansion is `sum([ColumnA]),sum([ColumnB]),sum([ColumnC])`. If the concatenation string is changed to a "+", then the axis would show the sum of the values from all columns included in the list: `sum([ColumnA])+sum([ColumnB])+sum([ColumnC])`.

For a categorical axis, such as the X-axis in a line chart or the category axis in a bar chart, you want to display a hierarchy rather than some calculated values. The map expression must then use angle brackets, "<>", and either NEST or CROSS to define what combinations to show. For example:

```
<$map("[${PropertyName}]", " NEST")>
```

See [Expression language details](#) on page 846 for more information about the NEST and CROSS alternatives.



As always when setting up expressions with multiple columns, you must make sure that the columns you use are of the same type and that they match the rest of the expression. For example, you cannot mix categorical and continuous columns on some axes, nor can you use categorical columns in an expression using any type of aggregation.

### More expression examples

If nothing else is stated, the expression examples below assume that you have a data table called "Data Table" containing three integer columns called "Column 1", "Column 2", and "Sales":

```
Column 1, Column 2, Sales
1,2,3
4,5,6
```

Requested result	Expression example	Resulting expression
Sum all integer columns in the data table called Data Table.	<code>\$map("sum([\$csearch([Data Table], "datatype:int")])", "+")</code>	<code>sum([Column 1])+sum([Column 2])+sum([Sales])</code>
Return a list of all columns in the data table called Data Table.	<code>\$csearch([Data Table], "**")</code>	<code>Column1,Column 2,Sales</code>
Return an escaped column name from a property (MyProperty) with the value "Column name with bracket ()".	<code>\$esc(\${MyProperty})</code>	<code>[Column name with bracket ()]</code>
Use a list-valued property expression as input to a data function.  In the example, the property MyListProperty contains three column names: Column 1, Column 2, and Sales.	<code>\$map("[Data Table].[\${MyListProperty}]", ",")</code>	<code>[Data Table].[Column 1],[Data Table].[Column 2],[Data Table].[Sales]</code>
Use multiple columns on an axis where one of the columns is retrieved via a property control.  In the examples, the property MyProperty has the value Column 2.	<code>[Column 1],[\${MyProperty}]</code> or <code>&lt;[Column 1] NEST [\${MyProperty}]&gt;</code> or <code>Sum([Column 1]), Sum(\${MyProperty})</code> etc.	<code>[Column 1],[Column 2]</code> or <code>&lt;[Column 1] NEST [Column 2]&gt;</code> or <code>Sum([Column 1]), Sum(Column 2)]</code> etc.

Requested result	Expression example	Resulting expression
Change the display name of multiple columns on an axis using a list-valued property.  In the example, the property MyListProperty contains three column names: Column 1, Column 2 and Sales.	<code>\$map("Sum([\${MyListProperty}]) as [\${MyListProperty}]", ",")</code>  All list-valued properties in the expression must be of the same size.	<code>Sum([Column 1]) as [Column 1], Sum([Column 2]) as [Column 2], Sum</code>

## Troubleshooting property expressions

Because properties can be inserted and interpreted in two different ways, there might be occasions where a seemingly correct expression does not work as expected.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client. You can write all of the expressions mentioned in the web client, but the installed client has some shortcuts and interfaces that are not available in the browser.

The following messages can be encountered when inserting properties in the **Expression** field of the Custom expression dialog or the Add calculated column dialog.

Some of the problems can also occur if you add a property to an axis using Set from Property and the expression does not match the axis. In that case, you must right-click and go to the **Custom expression** dialog to manually change your expression. Try to identify your problem using the table below.

Error text for the expression field	Expression example	Possible errors	Solution
The expression is not complete. or The expression is empty. or The expression cannot be evaluated.	<code>\${MyProperty}</code>	<p>When a property is inserted to an expression using double-click or by clicking on the <b>Insert Properties</b> button in the installed client, it is inserted as text.</p> <p>With this syntax, Spotfire will try to interpret a string property value as a column name or a part of an expression rather than as a value.</p>	<p>If you want to use the value of the string property, put quotation marks around the expression:  <code>"\${MyProperty}"</code></p> <p>You can also right-click on the property in the <b>Available properties for column list</b> (in the installed client) and select <b>Insert as Value</b> from the pop-up menu instead:</p> <pre>DocumentProperty("MyProperty")</pre> <p>If the property is supposed to hold a column name, but the column name contains space characters, you need to put "[" and "]" characters around the property expression. You can also use the <code>\$esc()</code> function that both escapes any "]" characters and converts the property string to a column:</p> <pre>\$esc("\${MyProperty}")</pre> <p>If the property is added using <b>Set from Property</b>, you will automatically get the escaped version of the expression.</p>

Error text for the expression field	Expression example	Possible errors	Solution
The expression is not complete.	<code>Concatenate("My first string", \${EmptyProperty})</code>	If a string property inserted as text is empty then it cannot be interpreted as a column and Spotfire will not see that the second argument in this example is there at all.	<p>If you want to use the value of the string property, put quotation marks around the expression:</p> <pre>Concatenate("My first string", "\${EmptyProperty}")</pre> <p>You can also right-click on the property in the <b>Available properties for column list</b> (in the installed client) and select <b>Insert as Value</b> from the pop-up menu instead:</p> <pre>Concatenate("My first string", DocumentProperty("EmptyProperty"))</pre> <p>If you want to use the content of a column in the concatenation, you should put '[' and ']' characters around the property to make sure the property is interpreted as a column (or use <code>\$esc()</code> as described above).</p> <pre>Concatenate("My first string", [\${EmptyProperty}])</pre>
Invalid type for function call 'DocumentProperty'	<code>DocumentProperty(MyProperty)</code>	<p>The property name should always be written within quotation marks when property functions are applied.</p> <p>Quotation marks are automatically added if you use the <b>Insert as Value</b> shortcut from the pop-up menu (in the installed client).</p>	<p>Put quotation marks around the property name:</p> <pre>DocumentProperty("MyProperty")</pre>

Error text for the expression field	Expression example	Possible errors	Solution
<p>Expected 'End of expression' but found ';' on line 1 character 12</p> <p>or</p> <p>The expression is not valid</p>	<pre>\$map( "sum([ \${MyListProperty} ])", " ", "</pre>	<p>When list-valued properties are used on an axis you must map the list-valued properties to a single string.</p> <p>The expression must contain a template to use for each value in the list (for example, an aggregation measure) and also a specification of how the list values should be connected in the resulting expression.</p> <p>The default expression obtained when first inserting the property is suitable for continuous axes where a simple listing of the columns included in the list-valued property is desired. In all other cases it must be manually modified.</p>	<p>Depending on what you want to display you must modify the default expression somewhat differently.</p> <p>If a simple listing of column names is desired (for example, if you want to show multiple columns on a bar chart value axis), then the expression in the example works fine. If you want to use an aggregation measure other than "sum" simply replace "sum" in the expression.</p> <p>If the list of columns is to be shown on a categorical axis, you must modify the default expression to something like this:</p> <pre>&lt;\$map( "[ \${MyListProperty} ]", "NEST" )&gt;</pre> <p>Categorical expressions must be surrounded by angle brackets, "&lt;&gt;", and you must also specify how different combinations of categories should be handled.</p> <p>Also, the current selection of columns in the list-valued property might be a mixture of continuous and categorical columns. Make sure that only columns of the same type are included in the property list.</p> <p>There are also more cases where the expression must be modified. See the section <i>List-valued properties (multiple select)</i> under <a href="#">Properties in expressions</a> on page 907 for more information.</p>
<p>Expected '.' but found '3' on line 1 character 5</p>	<pre>\${TimeSpanProperty}</pre>	<p>If you try to use a TimeSpan, Date, Time or DateTime property in an expression, the expression language will not be able to interpret it correctly without some manual editing.</p> <p>First, it needs quotation marks around the property name. This will interpret the property as a string and you will be able to get rid of the error. Second, you must use one of the conversion functions to actually interpret the value as a TimeSpan, Date, Time or DateTime.</p>	<p>Put quotation marks around the property name and use the corresponding conversion function:</p> <pre>TimeSpan( "\${TimeSpanProperty}" )</pre> <p>You can also right-click on the property in the <b>Available properties for column list</b> (in the installed client) and select <b>Insert as Value</b> from the pop-up menu instead:</p> <pre>TimeSpan( DocumentProperty( "TimeSpanProperty" ) )</pre>

Error text for the expression field	Expression example	Possible errors	Solution
{Table is undefined in \${Table}	<code>\${{Table}}.MyProperty</code>	If a name contains right curly bracket (}) it must be <b>escaped</b> by a backslash character (\).	To access the data table property MyProperty in a data table that is named "{Table}" you must write:  <code>\${{Table\}}.MyProperty</code>

## More about \$esc and \$csearch

### \$esc(expression)

The `$esc()` function is used to **escape** "]" which normally denotes the end of a column from column names and adds "[" and "]" to the column name. The brackets are required for column names containing space characters to be interpreted as columns. `$esc()` can be used together with the `$csearch()` function. The argument is an expression which could be a property value or a property function that starts with a dollar sign (\$).

For example, let the data table "A Data Table" have three columns called "Column 1", "Column 2[example]", and "Sales". The expression `$esc($csearch([A Data Table], "Col*"))` returns a list with two elements-the strings "[Column 1]" and "[Column 2[example]]".

### \$csearch([Data Table], "search string")

The `$csearch()` function is used to produce a "filtered" list of column names. It allows you to select a number of columns from a data table using a limiting search expression. This function is likely to be used together with the `$map()` function. The first argument is a data table and the second argument is a string that contains the search expression to be used to determine what column names should be returned. The function returns a list of (unescaped) column names contained in the data table that fulfills the search expression.

For example, let the data table "A Data Table" have three columns called "Column 1", "Column 2", and "Sales". The expression `$csearch([A Data Table], "Col*")` returns a list with two elements-the strings "Column 1" and "Column 2". If the property MyTable contains the string [A Data Table] and the property MyA contains the string "Col\*" then `$csearch(${MyTable}, "${MyA}")` will return the same result.

`$csearch` together with `$map()` and `$esc()` can produce column lists or calculations based on columns from list-valued properties. For example, `$map("sum($esc($csearch(${MyTable}, ""))", "+")` is expanded to `sum([Column 1])+sum([Column 2])+sum([Sales])` because the search expression `*` will return all columns in the table. The `$esc()` function is necessary if you want the strings in the list to be interpreted as columns and the column names contain space characters.



`$csearch()` is primarily intended to be used in visualization axis expressions or included in data function argument expressions. While `$csearch()` is looking at all columns in a data table, including any calculated columns, it is less suitable for use in calculated columns. If used in a calculated column, cyclic dependencies might occur.

## Data table or column names in preprocessor expressions

Because the preprocessor only does textual replacement without any context or knowledge of what is a column name and what is a number, there is no way to track when a column name has changed. This means that preprocessor expressions must be manually updated if such names change.

- If possible for your purpose, use non-preprocessor syntax instead of preprocessor syntax. For example, you can use `Integer(DocumentProperty("name"))` instead of `${name}` to convert the value to an integer, if the property is not already an integer.

- If that is impossible, perhaps you can use an intermediate column that does not get renamed? For example, assume you have an original column called A. You can then add a calculated column, B, with the expression [A]. Then you can mix B with a preprocessor syntax to calculate C. When replacing data, and A is renamed to A2, the B column will update correctly (because it does not have any preprocessor syntax in it), and C will recalculate correctly because it only depends on B, which was not renamed.

## Visualization properties in expressions

In some cases, you might want to use the expression or the display name currently set on an axis as a dynamic part of another expression, so that it changes with your selections. For example, this can be interesting in a visualization title or in a tooltip or label. You might also want one axis to be automatically configured to use the same expression as the one set on another axis. For example, you might want the color axis to automatically update to the same expression as what is used on a categorical axis.

There are a number of different "axis expressions working like properties" available in the visualizations. These properties exist only in the context of a visualization and they have values that are defined by the currently used settings in the visualization. For example, a scatter plot can expose the display name of its X and Y-axis expressions as properties and these can in turn be used to specify the visualization title.

The syntax to use is `${Axis.Axis Name.DisplayName}`, `${Axis.Axis Name.ShortDisplayName}` and `${Axis.Axis Name.Expression}`, for the display name, short display name and the underlying expression, respectively. For example, if the Axis Name is "X", the expression could refer to `${Axis.X.DisplayName}`, `${Axis.X.ShortDisplayName}`, or `${Axis.X.Expression}`. Note that the actual axis names to use can be different in two similar looking visualizations, see below. The `${DataTable.DisplayName}` property can be used in all visualizations to retrieve the name of the main data table used in the visualization.

These properties can be used the same way as other types of properties. See [Properties in expressions](#) on page 907 for more information. All of the display name-properties listed below also have a corresponding property for the actual expression.

Visualization	Axis name used in expression	Axis (property) it represents
[All visualizations]	<code>DataTable.DisplayName</code>	The name of the main data table used in the visualization
Cross table	<code>Axis.Columns</code> <code>Axis.Measures</code> <code>Axis.Rows</code>	Horizontal axis Cell values Vertical axis
Graphical table	<code>Axis.Rows</code>	Graphical table rows
Bar chart	<code>Axis.Color</code> <code>Axis.Columns</code> <code>Axis.Pages</code> <code>Axis.Panels</code> <code>Axis.Rows</code> <code>Axis.X</code> <code>Axis.Y</code>	Color by Trellis columns Trellis pages Trellis panels Trellis rows Category axis Value axis



Visualization	Axis name used in expression	Axis (property) it represents
Waterfall chart	Axis.Color	Color by
	Axis.Columns	Trellis columns
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rows	Trellis rows
	Axis.X	Category axis
	Axis.Y	Value axis
Line chart	Axis.Line	Line by
	Axis.Color	Color by
	Axis.Columns	Trellis columns
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rows	Trellis rows
	Axis.X	X-axis
	Axis.Y	Y-axis
Combination chart	Axis.Color	Color by
	Axis.Columns	Trellis columns
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rows	Trellis rows
	Axis.X	X-axis
	Axis.Y	Y-axis
Pie chart	Axis.Color	Color by
	Axis.Columns	Trellis columns
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rows	Trellis rows
	Axis.Size	Pie size by
	InnerAxis.Color	Color by
	InnerAxis.SectorSize	Sector size by

Visualization	Axis name used in expression	Axis (property) it represents
Scatter plot	Axis.Color	Color by
	Axis.Columns	Trellis columns
	Axis.DrawingOrder	Drawing order by
	Axis.Line	Connect by (draw a separate line per value in)
	Axis.Marker	Marker by
	Axis.Order	Order by (order each line by)
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rotation	Rotation by
	Axis.Rows	Trellis rows
	Axis.Shape	Shape by
	Axis.Size	Size by
	Axis.X	X-axis
	Axis.Y	Y-axis
	InnerAxis.Color	Pie sector color
	InnerAxis.SectorSize	Pie sector size
3D scatter plot	Axis.Color	Color by
	Axis.Columns	Trellis columns
	Axis.Marker	Marker by
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rows	Trellis rows
	Axis.Size	Marker size by
	Axis.X	X-axis
	Axis.Y	Y-axis
	Axis.Z	Z-axis
Map chart - Feature layer	Axis.Color	Color by
	Axis.Columns	Trellis columns
	Axis.Geometry	Feature by
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rows	Trellis rows

Visualization	Axis name used in expression	Axis (property) it represents
Map chart - Marker layer	Axis.Color	Color by
	Axis.Columns	Trellis columns
	Axis.DrawingOrder	Drawing order by
	Axis.Line	Connect by (draw a separate line per value in)
	Axis.Marker	Marker by
	Axis.Order	Order by (order each line by)
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rotation	Trellis rows
	Axis.Rows	Rotation by
	Axis.Shape	Shape by
	Axis.Size	Marker size by
	Axis.X	X-axis
	Axis.Y	Y-axis
	InnerAxis.Color	Pie sector color
	InnerAxis.SectorSize	Pie sector size
Treemap	Axis.Color	Color by
	Axis.Columns	Trellis columns
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rows	Trellis rows
	Axis.Size	Size by
	Axis.Hierarchy	Hierarchy
Heat map	Axis.Columns	Trellis columns
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rows	Trellis rows
	Axis.X	X-axis
	Axis.Y	Y-axis
	Axis.Measures	Cell values
KPI chart	Axis.Color	Color by
	Axis.Comparative	Comparative value
	Axis.Tile	Tile by
	Axis.X	X-axis
	Axis.Y	Y-axis
Parallel coordinate plot	Axis.Line	Line by
Summary table	Axis.CategoryAxis	Categorization

Visualization	Axis name used in expression	Axis (property) it represents
Box plot	Axis.Color	Color by
	Axis.Columns	Trellis columns
	Axis.Pages	Trellis pages
	Axis.Panels	Trellis panels
	Axis.Rows	Trellis rows
	Axis.X	X-axis
	Axis.Y	Y-axis

## Aggregations and statistical measures

In the visualizations and various tools, calculations are performed that use aggregation and statistical measures.

For a description of each measure, see the corresponding section.



For measures where a large statistical selection is needed, the result from a calculation may vary with the number of available values.

- [Sum](#)
- [Average](#) (Avg)
- [Count](#)
- [Unique Count](#)
- [Min](#)
- [Max](#)
- [Median](#)
- [Standard Deviation](#) (StdDev)
- [Standard Error](#) (StdErr)
- [Variance](#)
- [Lower endpoint of 95% Confidence Interval](#)
- [Upper endpoint of 95% Confidence Interval](#)
- [First Quartile](#) (Q1)
- [Third Quartile](#) (Q3)
- [Lower Adjacent Value](#) (LAV)
- [Upper Adjacent Value](#) (UAV)
- [CountBig](#)
- [Unique Concatenate](#)
- [Concatenate](#)
- [First](#)
- [GeoMetricMean](#)
- [Interquartile Range](#) (IQR)
- [Last](#)
- [Lower Inner Fence](#) (LIF)

- [Lower Outer Fence \(LOF\)](#)
- [Mean Deviation](#)
- [Median Absolute Deviation](#)
- [Most Common](#)
- [Outlier Count \(Outliers\)](#)
- [10th Percentile \(P10\)](#)
- [90th Percentile \(P90\)](#)
- [Outlier Percentage \(PctOutliers\)](#)
- [Product](#)
- [Range](#)
- [Upper Inner Fence \(UIF\)](#)
- [Upper Outer Fence \(UOF\)](#)

In the column selectors of some visualizations there are also a number of aggregation measures available that are in fact shortcuts to expressions. See below for a description:

- [Cumulative Sum](#)
- [Moving Average](#)
- [Difference](#)
- [Difference %](#)
- [Difference Year Over Year](#)
- [Difference % Year Over Year](#)
- [% of Total](#)
- [Year to Date Total](#)
- [Year to Date Growth](#)
- [Change Relative to Start](#)
- [Change Relative to Fixed Point](#)
- [Compound Annual Growth Rate](#)

See [Node navigation](#) and [Shortcuts to frequent custom expressions](#) for more general information about writing custom expressions with OVER and THEN.

## Sum and Product

Sum is the result of adding all values in a the subset. Product is the result of multiplying all values in the subset.

### Sum

If no categorization has been applied, then the sum will show the total sum of all values in each column:

$$\text{Sum} = x_1 + x_2 + x_3 \dots$$

## Product

If no categorization has been applied, then the product will show the total product of all values in each column:

$$\text{Product} = x_1 * x_2 * x_3 \dots$$

## Average

The average (Avg), or mean, is calculated as the sum of all values in the distribution divided by the number of values.

The arithmetic mean value,  $\bar{x}$ , is calculated as:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

## Count

Count gives the number of values in a column, not counting empty values. In the table below, Column A has a Count of 3, while Column B has a Count of 4.

Column A	Column B
1	4
	7
8	3
9	6

CountBig is used when the number of counted values is very large. It returns a LongInteger instead of an integer.

## Unique Count

Unique Count gives the number of unique (distinct) values in a column. Empty values are not counted. In the table below, column A has a Unique Count of two and column B has a Unique Count of three.

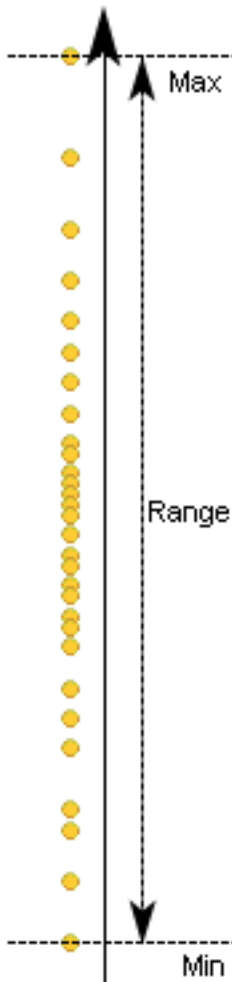
Column A	Column B
1	4
	4
1	3
9	6



In expressions, the DISTINCT keyword can be used to return a result using the unique values only. For example, Avg(DISTINCT[Column]) would return the average of the unique values rather than the average of all values in the specified column. UniqueCount([Column]) is the equivalent of Count(DISTINCT[Column]).

## Min and Max

Min is the minimum (lowest) value of the selected subset. Max is the maximum (highest) value of the selected subset. Range = Max-Min.



## Median

The median of a distribution is the value which, when the distribution is sorted, appears in the middle of the sorted list. If the number of values is even, the median is computed by taking the average of the two middle values.

The median is sometimes called the **location** of the distribution.

## Standard Deviation

The standard deviation (StdDev),  $s$ , is an indication of how dispersed the probability is about its center.

It is computed as follows:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

where

- $\bar{x}$  is the **average** value of the group
  - $n$  is the number of values in the group (Count)
- If the number of values ( $n$ ) is one, an error is returned.

## Standard Error

The standard error (StdErr) is the standard deviation of the calculated mean.

It is calculated as the estimated **standard deviation** divided by the square root of the size of the sample.

## Variance

The sample variance,  $s^2$ , is an indication of how dispersed the probability distribution is about its center.

It is calculated as follows:

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

where

- $\bar{x}$  is the **average** value of the group
  - $n$  is the number of values in the group (Count)
- If the number of values ( $n$ ) is one, an error is returned.

## Confidence Intervals

A confidence interval is the mean of an estimate plus and minus a variation in that estimate.

Confidence intervals are calculated as:

$$\bar{x} \pm \frac{1.959964 \times s}{\sqrt{n}}$$

where

- $\bar{x}$  is the **average** value of the group
- $s$  is the sample **standard deviation**
- $n$  is the number of values in the group (Count)

The L95 value is the lower endpoint of the confidence interval.

The U95 value is the upper endpoint of the confidence interval.

The static t-value of 1.959964, used by the L95 and U95 functions, is adapted for large sample sizes ( $n \geq 40$ ). For smaller sample sizes, use the following custom expressions instead:

L95: Avg([Value])-TInv(0.025,Count()-1)\*StdDev([Value])/Sqrt(Count())

U95: Avg([Value])+TInv(0.025,Count()-1)\*StdDev([Value])/Sqrt(Count())





## Percentiles and Quartiles

A percentile is a measure at which that percentage of the total values are the same as or below that measure. For example, 90% of the data values lie below the 90th percentile, whereas 10% of the data values lie below the 10th percentile.

Quartiles are values that divide a (part of a) data table into four groups containing an approximately equal number of observations. The total of 100% is split into four equal parts: 25%, 50%, 75% and 100%.

The first quartile (or lower quartile), Q1, is defined as the value that has an f-value equal to 0.25. This is the same thing as the twenty-fifth percentile. The third quartile (or upper quartile), Q3, has an f-value equal to 0.75. The interquartile range, IQR, is defined as Q3-Q1.

The percentiles and quartiles are calculated as follows:

1. The f-value of each value in the data table is computed:

$$f_i = \frac{i - 1}{n - 1}$$

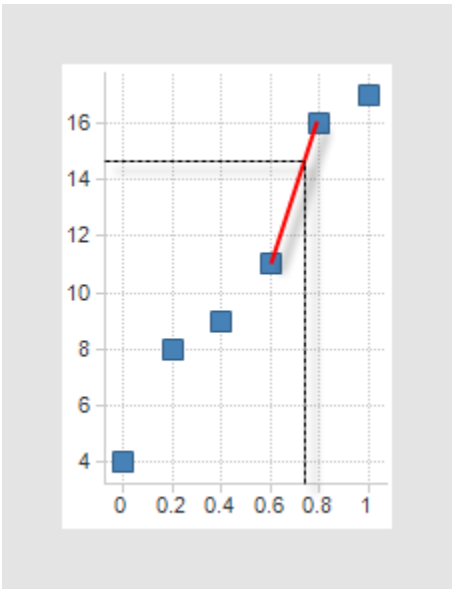
where  $i$  is the index of the value, and  $n$  the number of values.

2. The first quartile is computed by interpolating between the f-values immediately below and above 0.25, to arrive at the value corresponding to the f-value 0.25.
3. The third quartile is computed by interpolating between the f-values immediately below and above 0.75, to arrive at the value corresponding to the f-value 0.75.
4. Any other percentile is similarly calculated by interpolating between the appropriate values.

### Example

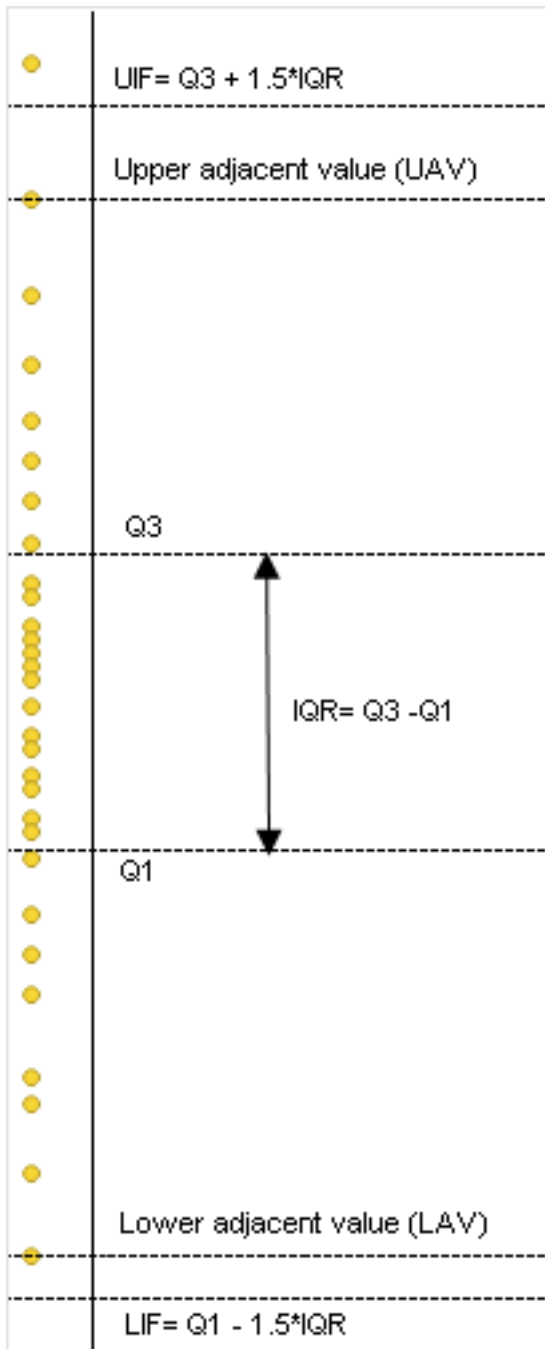
Value	f-value
4	0
8	0.2
9	0.4
11	0.6
16	0.8
17	1.0

Interpolation at f-value=0.75 yields Q3=14.75.



## Adjacent Values and Outliers

Upper and lower adjacent values can be calculated, and upper outer and lower outer fences that will define outlier values.



### Adjacent values

IQR above is the [interquartile range](#).

The upper adjacent value (UAV) is the largest observation that is less than or equal to the upper inner fence (UIF), which is the third [quartile](#) plus  $1.5 \cdot IQR$ .

The lower adjacent value (LAV) is the smallest observation that is greater than or equal to the lower inner fence (LIF), which is the first quartile minus  $1.5 \cdot IQR$ .



If, by the above definition, the UAV is such that it is smaller than Q3, then it is set equal to Q3. Similarly, the LAV is never allowed to be greater than Q1.

### Outliers

Outliers are all values that fall outside either of the fences. Outside values are values that fall in between the inner and outer fences. Far out values are outside the outer fence.

The upper outer fence (UOF) is defined as the threshold located at  $Q3 + (3 \times IQR)$ . The lower outer fence (LOF) is defined as the threshold located at  $Q1 - (3 \times IQR)$ .

The outlier percentage is the count of outliers divided by the total count for each category.

## Concatenate and Unique Concatenate

The Concatenate measure appends all values in the subset into a string, and the Unique Concatenate measure appends all the unique values into a string. That is, when you use Unique Concatenate, each value will only be included in the string once.

### Example

Column A	Column B	Column C
1	1	1
1		3
1	3	3

With the columns from the table above, the following result would be given:

Column	Concatenate	Unique Concatenate
A	1,1,1	1
B	1,3	1,3
C	1,3,3	1,3

## First and Last

The First and Last measures address values based on the physical order of the data rows.

First returns the first valid value in the selected column, and Last returns the last valid value.

## Geometric Mean

The geometric mean is a mean that multiplies all values and calculates a root of the result.

The geometric mean value,  $G$ , is calculated as:

$$G = \sqrt[n]{x_1 x_2 \dots x_n}$$

That is, it is the  $n^{\text{th}}$  root of the product of all values in a distribution, where  $n$  is the number of values. It is generally preferred over the arithmetic mean when working with normalized values, percentages or exponential values. It is not suitable for calculations including any negative values or zeros.

## Mean Deviation

The mean deviation (average absolute deviation, AAD) is calculated as the mean of the absolute difference between a value and the mean value of the group:

$$\text{Mean Deviation} = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$$

where

- $\bar{x}$  is the [average](#) value of the group
- $n$  is the number of values in the group (Count)

## Median Absolute Deviation

The median absolute deviation (MAD) is calculated as the median of the absolute value of each value,  $x_i$ , minus the median of  $x$ :

$$MAD = \text{median}(|x_i - \tilde{x}|)$$

where  $x_i$

- $\tilde{x}$  is the median value of the group

The median absolute deviation is used instead of the mean deviation when the deviation value needs to be less affected by extreme values in the tail. This is due to the fact that the median is less affected by the tail values than the mean is.

## Most Common

The Most Common measure returns the most common value of the selected column.

If several values are equally common, the first one will be used.

# Exporting your analysis

You can export your analysis, or data used in the analysis, to different file formats and locations to share it with others.

When you want to share your discoveries in an analysis with others, you can export the entire analysis, or parts of it, to PDF or to Microsoft PowerPoint. You can also export separate visualizations to disk and save them as images. The data that is used in the analysis can be exported to the library.

## Exporting to PDF

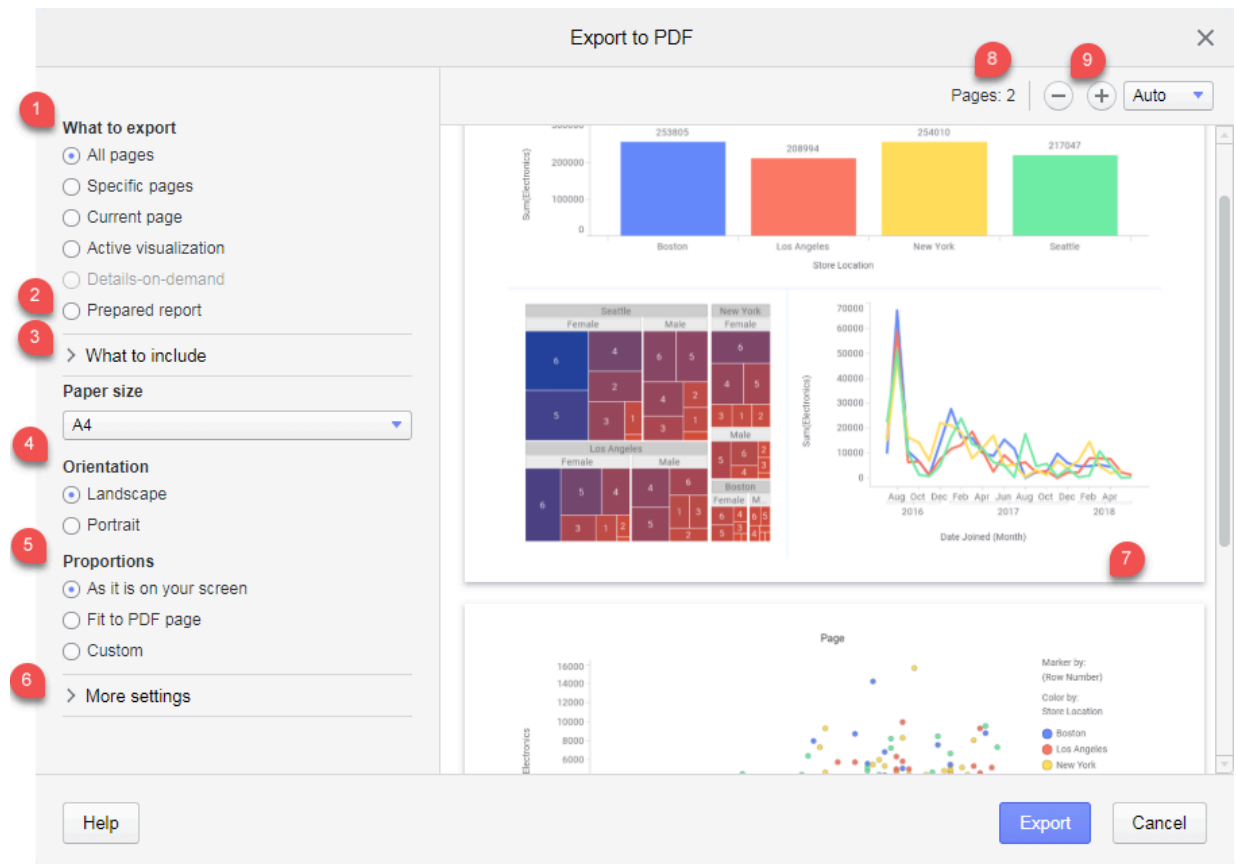
You can export what is visualized in an analysis to a PDF document using the Export to PDF dialog.

The dialog is accessed by selecting **File > Export > To PDF** on the menu bar.

From the dialog, you can

- do an ad-hoc, basic export of the content in an analysis, and get a nice PDF output.
- prepare customized [reports](#) to be exported. When you prepare such reports, you have access to export settings that are more advanced. You can, for example, apply bookmarks and filters in the report to mirror certain interactivity in the analysis.

In both cases, a preview of the PDF output is provided.



An example of a Export to PDF dialog is shown above.

Beneath **What to Export** (1), you can select to export the entire analysis (**All pages**), parts of it, or prepared analysis reports (2). A simple click on **Export** in the lower right corner returns the PDF output. You can, though, before clicking **Export**, modify the appearance of the PDF output using various settings. Which modifications you can do depend on what you select to export.



When exporting a prepared report, all settings are made within the report itself. You only need to click **Export** to create the PDF document. See [Prepared reports](#) on page 938 for more information.

The lower left part of the dialog contains settings that are general to the different **What to export** options (**Prepared report** is an exception). You can

- beneath **What to include** (3), decide whether page titles, page numbers, date, and annotations should be included in the export.
- define the **Paper size** and page **Orientation** (4), and customize the **Proportions** of the content on a page (5). The option **As it is on your screen** keeps the proportions used in the analysis. It is, however, possible to change the Aspect ratio, that is, the relations between width and height of the content of the pages using the **Custom** option. If you want to directly adapt the proportions so the entire PDF page is utilized, the option **Fit to PDF page** is available.

When you specify other proportions than are used in the analysis itself, you can also adjust the size of the texts on the page. This can be useful, for example, when the preview shows that some visualization text is truncated. If you decrease the text size, the text can be fully visible.

- beneath **More settings** (6), specify the margins to the edge of the PDF pages and the quality of the images in dots per inch (dpi).



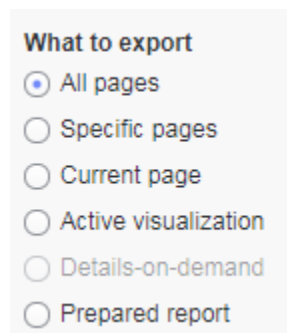
Specifying more dots per inch results in larger files sizes.

The preview (7) represents the final PDF pages. It reflects what is currently selected to be exported, and the settings made. Above the preview, the number of PDF pages that are created in the export is displayed (8), and you can use the scroll bar to navigate among them. Buttons for zooming in and out in the preview are available (9). By zooming out, for example, you get an overview of the pages.

## What to export

The entire analysis, parts of it, and reports can be exported to a PDF document.

The selection of what to export is made in the upper-left corner of the [Export to PDF](#) dialog.



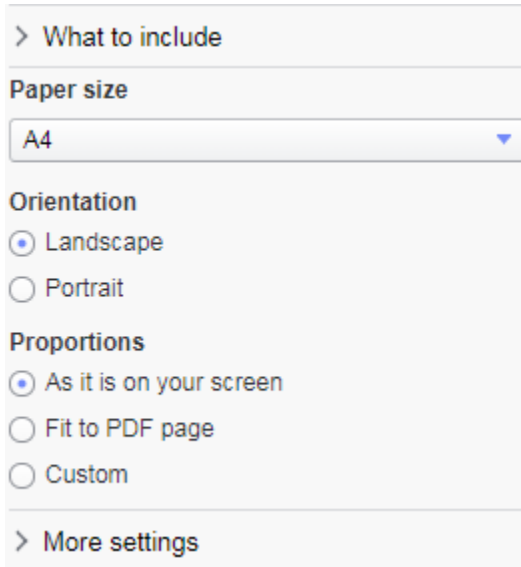
You select

- **All pages** to [export the entire analysis](#).
- **Specific pages** to [export a selection of the pages](#) in the analysis.
- **Current page** to [export a single analysis page](#).
- **Active visualization** to export a single visualization. Subject to the type and current setup of the visualization you want to export, you can
  - [export what is currently visible in the visualization](#).
  - [for a visualization with not all content visible, in the export include also what is not visible](#).

- **Details-on-Demand** to [export the data rows that underlie visualization items that are marked](#). Also here you can export only what is currently visible, or let the export also comprise content that is not visible in the Details-on-Demand panel. See also [Displaying item details](#) on page 596.
- **Prepared report** to [export analysis reports](#) that have been prepared in advance.

## General export settings

What to export is selected in the upper-left part of the Export to PDF dialog, and a selection is associated with specific settings. The lower-left part of the dialog contains settings that are general for all kinds of export selections. For example, you can specify page orientation and whether date and page numbers should be included in the PDF document.



> What to include

**Paper size**

A4 ▼

**Orientation**

☒ Landscape

☐ Portrait

**Proportions**

☒ As it is on your screen

☐ Fit to PDF page

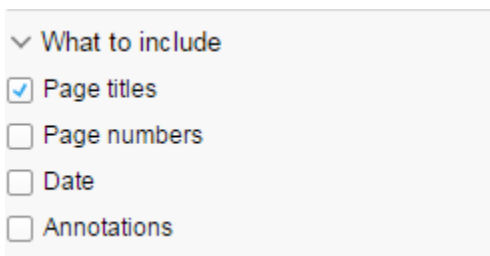
☐ Custom

> More settings

The preview in the dialog updates dynamically to reflect the latest settings.

### What to include

Beneath **What to include**, select whether or not **Page titles**, **Page numbers**, **Date**, and any added [Annotations](#) should be included in the PDF document.



▼ What to include

☒ Page titles

☐ Page numbers

☐ Date

☐ Annotations

### Paper size

In the drop-down list, select the paper size of the PDF document: **A4**, **Letter**, **Legal**, **A3**, or **A5**.

### Orientation

Click **Landscape** or **Portrait** to select page orientation in the PDF document.

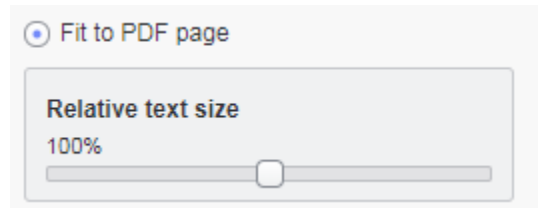


## Proportions

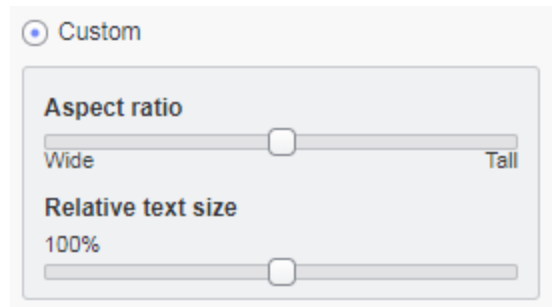
For specifying proportions of the content area on the PDF pages, the following options are available:

☒ As it is on your screen

Select **As it is on your screen** to, in the PDF document, keep the proportions used in the analysis. If you resize the content in the analysis, it will be resized in the PDF document as well.



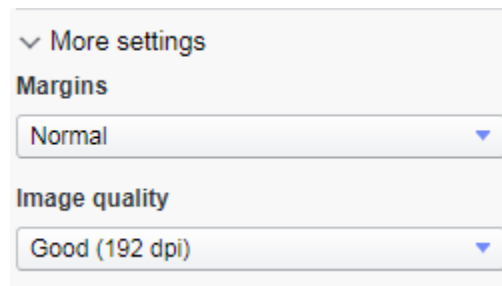
Select **Fit to PDF page** to apply the aspect ratio that will fill the entire space of the PDF page. This means that the proportions of the content in the PDF document will be different from the proportions in the analysis. If you want, drag the slider to adjust the **Relative text size** in the visualizations. Moving the slider furthest to the right makes the texts twice as big, and moving the slider furthest to the left half as big.



Select **Custom** to tune the proportions of the content area on the PDF page. The **Aspect ratio** slider is used to change the proportional relationship between width and height of the area. You can also drag the **Relative text size** slider to adjust the text sizes in the visualizations. Moving the slider furthest to the right makes the texts twice as big, and moving the slider furthest to the left makes the texts half as big.

## More settings

Beneath **More settings**, specify the size of the **Margins** outside the actual content on the PDF pages. The options in the drop-down list are **None**, **Narrow**, **Normal**, and **Wide**.



If needed, change the raster graphics quality by selecting the option containing the wanted number of dpi (dots per inch) in the **Image quality** drop-down list.



An increase of the dpi value will increase the file size.

## Exporting an entire analysis

You can export the entire analysis to a PDF document. Each page in the analysis corresponds to a page in the created PDF document.

### Procedure

1. On the menu bar, select **File > Export > To PDF**.  
The Export to PDF dialog is displayed.
2. Beneath **What to export**, make sure **All pages** is selected.
3. If you want, use the general [settings](#) in the dialog to modify the PDF output.
4. Click **Export**.

### Result

The created PDF document is downloaded to your computer.

## Exporting a visualization

You can do a direct export of a visualization to a PDF document. The document will consist of a PDF page that contains what you currently see in the visualization.

In most cases, all the content in a visualization is visible. The following procedure describes how to export such a visualization. The content on the created PDF page will reflect what is currently visible in the analysis. You can, however, still decide whether or not the visualization title, the legend, and the axis selectors should be visible on the PDF page. A preview of the PDF output will always be provided.

In case the entire visualization is **not** visible in the analysis, and for that reason a scroll bar is added to the visualization, you have two options:

- In the export include only what is visible following the procedure below.
- [In the export include also what is not visible.](#)

### Procedure

1. In the analysis, click the visualization that you want to export to a PDF document.  
The visualization becomes the active visualization.
2. Right-click the visualization, and select **Export > Visualization to PDF**.  
The Export to PDF dialog is displayed.  
  
Alternatively, you can, to the left on the menu bar, select **File > Export > Visualization to PDF**.
3. Beneath **What to export**, make sure **Active visualization** is selected.
4. To **Show visualization title**, **Show legend**, or **Show axis selectors**, make sure corresponding check boxes are selected.
5. For a tabular visualization, the **Export entire table** check box is provided. Make sure the check box is clear.
6. If you want, use the general [settings](#) in the dialog to modify the PDF output.
7. Click **Export**.

### Result

The created PDF document is downloaded to your computer.

## Exporting a visualization and include non-visible content

You can export a visualization to a PDF document. Sometimes the content in a visualization is extensive, and a scroll bar is added to make it accessible. When you export such a visualization, it is possible to export the entire content, that is, include also the content that is currently non-visible.

To export only what is visible in the visualization, see [Exporting a visualization](#) on page 934.

Visualizations whose entire content might not be visible are visualizations [trellised by page](#), or tabular visualizations (table, cross table, summary table, and graphical table).



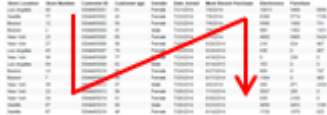
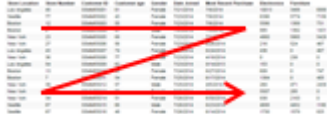
When exporting a [KPI chart](#), you can only export what is currently visible in the analysis.

### Procedure

1. In the analysis, click the visualization that you want to export.  
The visualization becomes the active visualization.
2. Right-click the visualization, and select **Export > Visualization to PDF**.  
The Export to PDF dialog is displayed.

Alternatively, you can, to the left on the menu bar, select **File > Export > Visualization to PDF**.

3. Beneath **What to export**, make sure **Active visualization** is selected.
4. To export also what is not visible in the visualization, select the **Export all trellis pages** check box for trellised visualizations, or the **Export entire table** check box for tabular visualizations.  
Depending on the type of visualization, and its current configuration, you encounter some of the check boxes listed below:

Check box	Description
<b>Show visualization title</b>	Specify whether or not the visualization title should be included in the export.
<b>Show legend</b>	Specify whether or not the legend should be included in the export. Consider clearing the check box if the legend information is redundant, or if more space is wanted.
<b>Show axis selectors</b>	Select this check box to include all axis selectors in the export, or clear it to hide them. If the axis selectors do not add value to the visualization, consider hiding them to free more space.
<b>Columns first, then rows</b>	<p>Tabular visualizations that fill up more than one PDF page are by default exported so that rows are exported first and then the columns:</p>  <p>Select this check box to instead export the columns first and then the rows:</p> 
<b>Repeat column headers</b>	For tabular visualizations, select this check box to display column headers on each PDF page on which the visualization continues.

Check box	Description
<b>Repeat row headers</b>	For cross tables, select this check box to display row headers on each PDF page on which the visualization continues.
<b>Keep hierarchy levels together</b>	Select this check box to, as far as possible, display items on the lowest level in a hierarchy on the same PDF page.
<b>Repeat frozen columns</b>	If columns are frozen in a table visualization, select this check box to display these columns on every PDF page on which the table visualization continues.

5. If you want, use the general [settings](#) in the dialog to modify the PDF output.
6. Click **Export**.

### Result

The created PDF document is downloaded to your computer.

## Exporting a single analysis page

You can export a single analysis page to a PDF document.

See also [Exporting an entire analysis](#) and [Exporting specific analysis pages](#) on page 936.

### Procedure

1. Click the page tab, whose content you want to export.  
The page becomes the active page, which is indicated by a thin colored line at the bottom edge of the tab.
2. On the menu bar, select **File > Export > Export to PDF**.  
The Export to PDF dialog is displayed.
3. Beneath **What to export**, make sure **Current page** is selected.  
If the analysis page contains more than one visualization, it is possible to export the visualizations to separate pages. Select the **Create one PDF page per visualization** check box.
4. If you want, use the general [settings](#) in the dialog to modify the PDF output.
5. Click **Export**.

### Result

The created PDF document is downloaded to your computer.

## Exporting specific analysis pages

You can export a selection of the analysis pages to a PDF document.

To export all pages, see [Exporting an entire analysis](#) on page 934, and to export the page that is currently visible in the analysis, see [Exporting a single analysis page](#) on page 936.

### Procedure

1. On the menu bar, select **File > Export > Export to PDF**.  
The Export to PDF dialog is displayed.
2. Beneath **What to export**, select **Specific pages**.

3. In the text field, type the numbers of the pages and/or page ranges that you want to include in the PDF document, for example, 4, 5, 7-10, and press Enter. The page tab furthest to the left in the analysis is page 1, the second left is page 2, and so on.



It is possible to export only a single page by typing its number.

4. If you want, use the general [settings](#) in the dialog to modify the PDF output.
5. Click **Export**.

### Result

The created PDF document is downloaded to your computer.

## Exporting Details-on-demand

In a visualization, you can mark items to investigate them further. For example, in the Details-on-demand panel, you can view the data rows underlying what is marked. The content in this panel can be exported to a PDF document.

For information about the Details-on-demand panel, see [Displaying item details](#) on page 596.

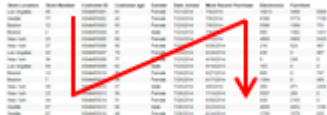
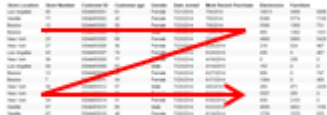
### Procedure

1. If the Details-on-demand panel is not visible on the current analysis page, select **View > Details-on-demand** on the menu bar.  
The Details-on-demand panel is displayed on the right-hand side of the page.
2. In a visualization, [mark the items](#) of interest.  
The data rows included in the marked items are listed in the Details-on-demand panel.
3. On the menu bar, select **File > Export > To PDF**.  
The Export to PDF dialog is displayed.
4. Beneath **What to export**, select **Details-on-demand**.



Often all content is not visible in the Details-on-demand panel, so a horizontal and/or vertical scroll bar are added to the panel. The content that is not visible can be included in the export as well.

5. To export also what is not visible, select the **Export entire table** check box.  
The following check boxes are displayed:

Check box	Description
<b>Columns first, then rows</b>	<p>Details-on-demand content that fills up more than one PDF page is by default exported so that rows are exported first and then the columns:</p>  <p>Select this check box to instead export the columns first and then the rows:</p> 

Check box	Description
<b>Repeat column headers</b>	Select this check box to display column headers on each PDF page on which the Details-on-demand content continues.
<b>Repeat frozen columns</b> (only available if a number of frozen columns has been specified for the Details-on-demand table in the installed client)	Select this check box to display the frozen columns on every PDF page on which the Details-on-demand content continues.

6. If you want, use the general [settings](#) in the dialog to modify the PDF output.
7. Click **Export**.

### Result

The created PDF document is downloaded to your computer.

## Exporting a prepared report

Reports of an analysis can be prepared in advance, and then exported to a PDF document.

See [Prepared reports](#) on page 938 for more information.

### Procedure

1. On the menu bar, select **File > Export > To PDF**.  
The Export to PDF dialog is displayed.
2. Beneath **What to export**, make sure **Prepared report** is selected.
3. In the opened drop-down list, select the report you want to export.
4. Click **Export**.

### Result

The PDF report is downloaded to your computer.

## Prepared reports

Reports of an analysis can be prepared in advance. Then the reports can be exported directly into a PDF document.

In a report, the content and the appearance can be customized in more detail than is possible using the other PDF export options. You can, for instance:

- Mix all or selected analysis pages with individual visualizations in a single document. If an individual, tabular visualization includes non-visible content, you have the possibility to display all its content in the report. This is also the case for visualizations trellised by page.
- Arrange the order of the PDF pages.
- Specify whether any filtering and marking made in the analysis should be considered in the report.
- Export different views of an analysis, where each view reflects a certain category in the data. This is done by filtering to one category at a time in a column.
- Export different views of the analysis based on analysis states that are captured in bookmarks. The views can, for example, show where marked items appear on various analysis pages, or certain interesting findings in the analysis.

- Customize contents and styling of the header and footer on the PDF pages.

You access the report settings by selecting **Prepared report** in the Export to PDF dialog, and clicking **New/Edit**. Because the report is saved as a part of the analysis, it is available to anyone that has access to the analysis.

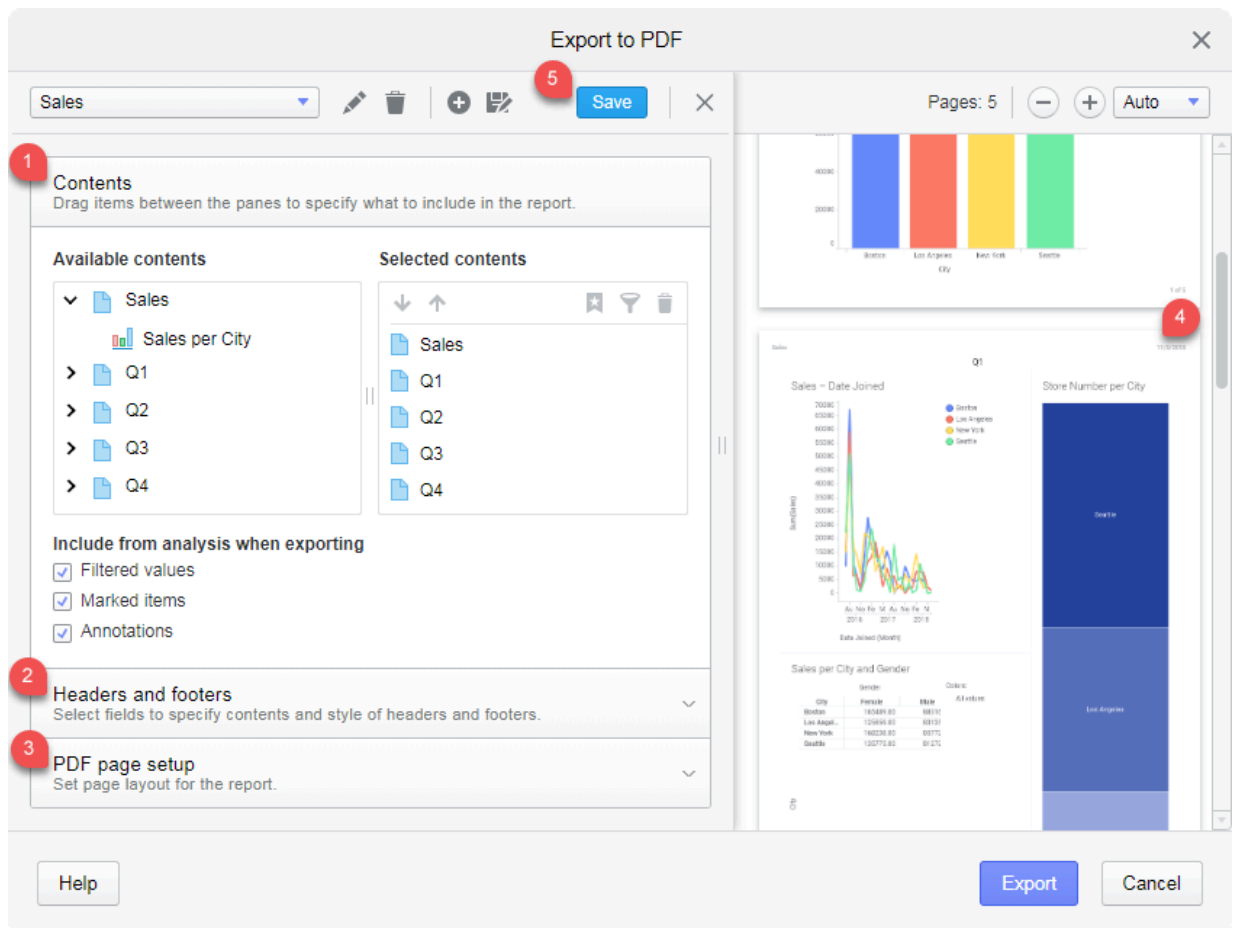


When you [export a report](#), all settings are already included, and no further adjustments are made. You simply click the **Export** button.

## Report settings

Reports of an analysis can be prepared in advance, and then exported to a PDF document. An overview of the report settings is shown below.

You access the settings by selecting **Prepared report** in the Export to PDF dialog, and then clicking **New**. For existing reports, select the report and click **Edit**.



The main parts of the report settings are

- **Contents** (1), where you decide which analysis pages should be included in the report, and which individual visualizations. This part is also where you find settings for whether or not to consider any filtering and marking made in the analysis, and settings for export of different views of the analysis.
- **Headers and footers** (2), where you specify what to display at the top and bottom of the PDF pages such as report title and page numbers. You can also put your own visual style on the headers and footers, for example, you can change fonts and colors.
- **PDF page setup** (3), where you set paper size, page orientation, and proportions of the content.

The preview reflects your current settings (4). For performance reasons, the preview might not be available directly. Then you launch it by clicking its **Update preview** button.

A report is saved as a part of the analysis, and it is available to anyone that has access to the analysis.

You must save the report to keep your report settings changes, and you can anytime click **Save** (5) while working on it.

## Contents

In the **Contents** part of the report settings, you define what in the analysis to include in the report. It is possible to mix all or selected analysis pages with individual visualizations.

The **Available contents** pane lists all the analysis pages and their visualizations, and the **Selected contents** pane shows what to be exported. You simply drag the items between the **Available contents** and **Selected contents** panes to specify the report content. If you want, you can re-order the items in the **Selected contents** pane afterwards by dragging them to the wanted position in the report.



It is possible to include also [hidden pages](#) in a prepared report. A dimmed page icon indicates the page has been hidden.

### Contents

Drag items between the panes to specify what to include in the report.

#### Available contents

▼

P1
 

Vis A
 Vis B

▼

P2
 

Vis C
 Vis D

#### Selected contents

↓

↑

P1
 P2

#### Include from analysis when exporting

☒ Filtered values
 ☒ Marked items
 ☒ Annotations

Moreover, you can use the check boxes **Filtered values** and **Marked items** to specify whether any filtering and marking made in the analysis should be included in the report. **Annotations** that have been added to the analysis can be shown in the report as well.

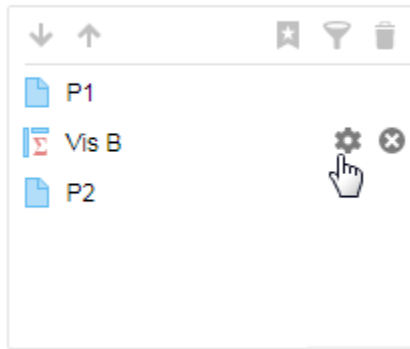
## Individual visualizations

Typically, each item in the **Selected contents** pane is a page of its own in the PDF document. However, for an individual visualization that has been added to the report, it is possible to make modifications to the output. What you can do depends on the type of the visualization. For example, if the visualization content is not fully visible in the analysis (this might be the case for trellised and tabular visualizations), you can specify to include also those non-visible parts in the report.

Spotfire® Web Client User Guide



### Selected contents



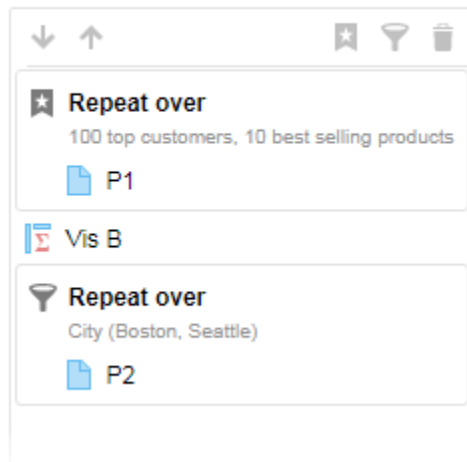
To access the modification options, you move the cursor over the added visualization, and click . The various options are described in [Exporting a visualization and include non-visible content](#) on page 935.

### Export different views of an analysis

It is possible to include different views of the analysis in the report without having to create each of them manually. The differences in the views can be based on bookmarks that have been applied, or based on filtering to selected filter values. By repeating pages and visualizations over bookmarks or filter values, you are able to mirror certain analysis interactivity in the report.

In the **Selected contents** pane, you specify which pages and visualizations you want to repeat the bookmarks or filter values over. What to repeat is gathered into framed sections. The export will then automatically return the different views of the analysis.

### Selected contents



Analysis pages and visualizations that are not included in any frame are not repeated.

### Export different views of an analysis using bookmarks

Assume you have found certain interesting analysis states and captured them in different bookmarks. You want to create a report that shows the pages and visualizations in these different states. Instead of manually applying each bookmark and exporting once for each applied bookmark, you can in a report apply the bookmarks to the pages in one single operation. You select which parts of the analysis you want different views of, followed by which bookmarks you want to apply, and the export will automatically be repeated over these bookmarks. See [Repeating pages and visualizations over bookmarks](#) on page 946.

## Export different views of an analysis using filters

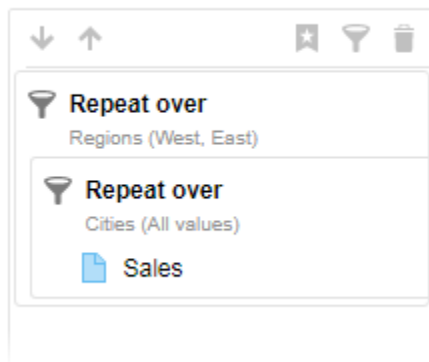
Perhaps you have an analysis with sales data for the 'Regions' West, East, North, and South, and you want to create a report that shows sales for only one region at a time. Instead of manually adjusting the 'Regions' filter and exporting once for West, once for East, and so on, you can make the export in one single operation. You select which parts of the analysis you want different views of, followed by which regions you are interested in, and the export will automatically be repeated over these regions, filtering to one region at a time. See [Repeating pages and visualizations over filter values](#) on page 947.

## Nested sections

More advanced repetitions can be achieved by creating repeating sections within other repeating sections. This can be useful if you, for example, want a hierarchically structured report.

To exemplify, assume your data (in addition to 'Regions' mentioned above) also contains 'Cities' such as Seattle, Los Angeles, Boston, and so on. Then you can, by nesting sections that repeat over filter values, get PDF pages that show combinations of categories like "West, Seattle", "West, Los Angeles", "East, Boston", and so on.

### Selected contents



## Headers and footers

In the **Headers and footers** part of the report settings, you specify what to display at the top and bottom of the pages exported to PDF.

The headers and footers of the PDF pages are divided into fields, which you can use to display details about the PDF document such as titles, page numbers, date, and any filter values and bookmarks that are applied. You can also change the visual appearance of the fields.

### Headers and footers

Select fields to specify contents and style of headers and footers.

Report title

Date

Page title

Page number of total

**What to display**

Page title

`${Report.PageTitle}`

**Font**

Roboto

16

**B**
*I*
≡
≡
≡
≡

**Background**

☐ None  
☒ Solid

**Borders**

☐ None  
☒ Solid

**Border width**

Top	Right	Bottom	Left	
0	0	0	0	<

**Border color**

Top	Right	Bottom	Left	
				<

Reset

In the symbolic presentation of the PDF pages, you simply click the field you want to make settings for:

- In the **What to display** drop-down list, you select what kind of information to display in the field. Because the content you insert in a field is of dynamic character, it is represented by a property. The value of such a property will then change to reflect the current circumstance on the separate PDF pages. The property name of the option you select is displayed in the text field. If you want, you can type your own text to display along with the properties, or to display stand-alone.
- Beneath **Font**, you specify the appearance of the text, and its horizontal and vertical alignment within the field. The selector includes a list of preferred fonts, which can be defined by the Spotfire

administrator, to help ensuring that an analysis will look the same for everybody that uses the analysis, including those who are viewing the analysis with web clients.



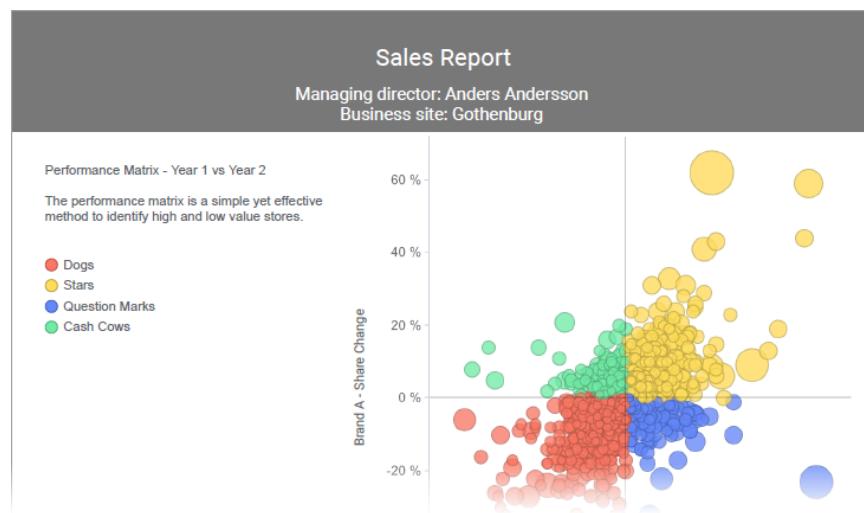
If you use a font that is not available on the Web Player or Automation Services service host, other users might not get the same experience when looking at the analysis as you do. Fonts might be replaced with other fonts. This is particularly important if the node manager for the service is installed on a [Linux host](#), because the fonts on Linux often differ from Windows fonts. If you want to use a specific font that is currently not included in the **Preferred fonts** section, you might want to discuss with an administrator whether the font can be installed on the node manager host and added as a PreferredFonts preference.

- Using the **Background** and **Borders** settings, you can color the background and borders of the field, and set the border widths.



You can click **Reset** to return to the default settings for the selected field.

### Example of a header



In the header above, the three top horizontal fields are given the same background color. The height of the top fields adapts to the height of the typed text.



To make settings for multiple fields at a time, press **Ctrl + click** the fields of interest.

## PDF page setup

In the PDF page setup part of the report settings, you specify paper size, page orientation, and proportions of the content.

**PDF page setup**  
Set page layout for the report.

**Paper size**  
A4

**Orientation**  
☐ Landscape  
☒ Portrait

**Proportions**  
☐ As it is on your screen  
☒ Fit to PDF page

**Relative text size**  
 100%  
 [Slider bar]  
☐ Custom

**Margins**  
Normal

**Image quality**  
Good (192 dpi)

For more information, see [General export settings](#) on page 932.

## Preparing a report

Reports of an analysis can be prepared in advance, and then exported to a PDF document.

The content of a report, and its appearance, can be customized using different report settings. Because all report settings are saved within the report itself, you only need to click **Export** to create the PDF document; no further adjustments are made.



While working on the report, the preview will reflect your settings. For performance reasons, the preview might not be available until you click **Update preview** in the preview area.

### Procedure


1. On the menu bar, select **File > Export > To PDF**.  
The Export to PDF dialog is displayed.
2. Beneath **What to export**, make sure **Prepared report** is selected.
3. Click **New**.  
The [report settings](#) are available.

4. Give the report a **Title**, and type a **Description** (optional).
5. Beneath **Contents**, make sure the analysis pages and individual visualizations you want to include in the report are listed in the **Selected contents** pane. You include, or exclude, items by dragging them between the **Available contents** and **Selected contents** panes.

Initially, all pages in the analysis are listed in the **Selected contents** pane.



You can re-order the items in the **Selected contents** pane by dragging them to the wanted positions.

6. For an individual visualization that is added to the **Selected contents** pane, move the cursor over it, and click . Then make the settings that are specific for the type of visualization. For information about the settings, see [Exporting a visualization and include non-visible content](#) on page 935.
7. If you want to export different views of the analysis, that is, repeat pages and visualizations over bookmarks or filter values, follow the instructions in [Repeating pages and visualizations over bookmarks](#) on page 946 and [Repeating pages and visualizations over filter values](#) on page 947.
8. Specify, using the check boxes at the bottom of **Contents**, whether or not **Filtered values**, **Marked items**, and **Annotations** from the analysis should be included in the report.
9. Specify what content to display in **Headers and footers**, and put your visual style to them.
10. Beneath **PDF page setup**, specify [settings related to the layout](#) of the report.
11. Click **Save**.

## Result

The prepared report is available for selection beneath **What to export** in the Export to PDF dialog.

## Repeating pages and visualizations over bookmarks

In a report, you can export different views of the analysis, where the views show the states of selected analysis pages and individual visualizations after applying different bookmarks. The different views are achieved by repeating selected pages and visualizations over the bookmarks.


For more information, see [Contents](#) on page 940.

## Prerequisites

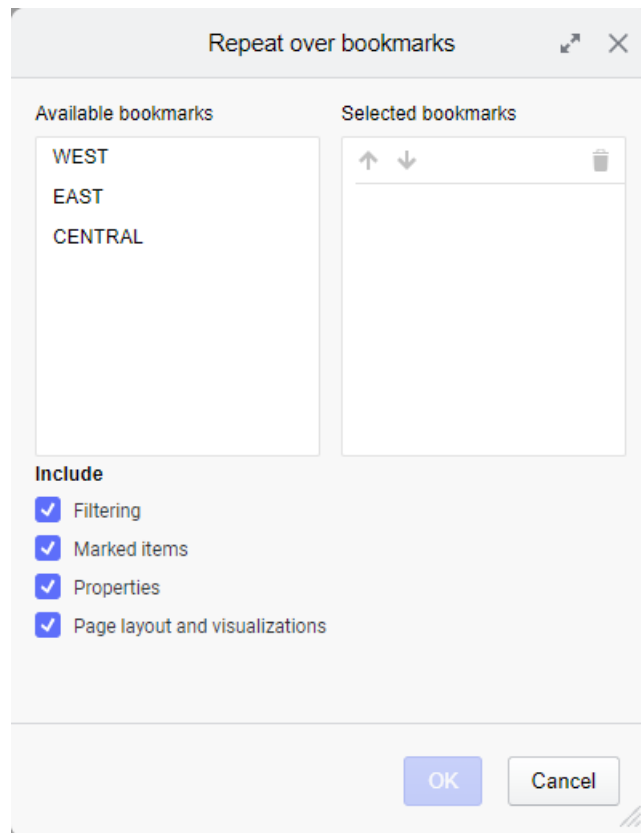
The analysis pages and individual visualizations you want different views of are listed in the **Selected contents** pane. The bookmarks you want to repeat pages and visualizations over are created.

## Procedure

1. In the **Selected contents** pane, select the analysis pages and individual visualizations you want different views of in the export.

2. Click  in the upper right of **Selected contents**.

In the **Selected contents** pane, a thin frame surrounds the section showing what to be exported in different views. The Repeat over bookmarks dialog opens.



3. Drag the bookmarks, which you want to apply to the section items, from the **Available bookmarks** pane to the **Selected bookmarks** pane.
4. Select which parts of the selected bookmarks you want to apply to the section items; **Filtering**, **Marked items**, **Properties**, and **Page layout and visualizations**.
5. Click **OK**.  
The dialog closes.
6. Click **Save**.

### Repeating pages and visualizations over filter values

In a report, you can export different views of the analysis, where each view reflects a certain category in a data column. This is done by an automatic, repeated filtering to one category at a time on selected pages and visualizations.


For more information, see [Contents](#) on page 940.

#### Prerequisites

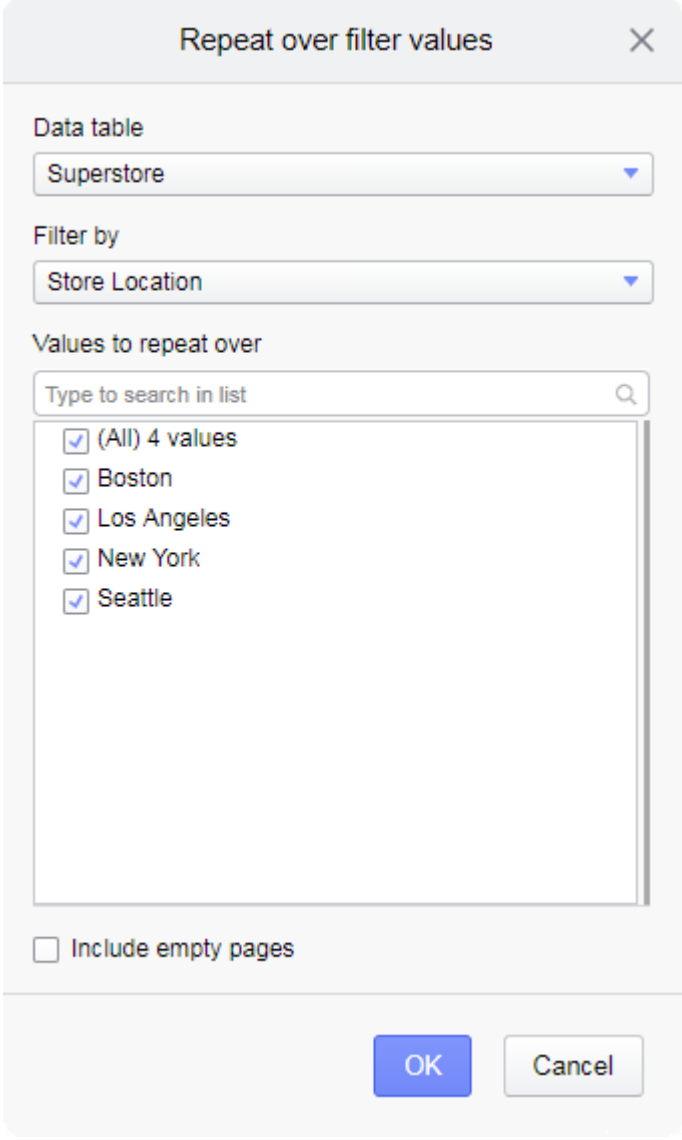
The analysis pages and individual visualizations you want different views of are listed in the **Selected contents** pane.

#### Procedure

1. In the **Selected contents** pane, select the analysis pages and individual visualizations you want different views of in the export.

2. Click  in the upper right of **Selected contents**.

In the **Selected contents** pane, a thin frame surrounds the section showing what to be exported in different views. The Repeat over filter values dialog opens.



The dialog box is titled "Repeat over filter values" with a close button (X) in the top right corner. It contains the following sections:

- Data table:** A dropdown menu showing "Superstore".
- Filter by:** A dropdown menu showing "Store Location".
- Values to repeat over:** A list box with a search bar at the top labeled "Type to search in list". The list contains five items, each with a checked checkbox: "(All) 4 values", "Boston", "Los Angeles", "New York", and "Seattle".
- Include empty pages:** An unchecked checkbox at the bottom left.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

3. If you have more than one data table in the analysis, select the **Data table** that contains the column (or hierarchy), whose categories you are interested in. (This setting is not visible for analyses based on a single data table.)
4. Beneath **Filter by**, select the column or hierarchy that contains the categories you want to filter to. Only columns containing categorical values are available for selection. The categories you can filter to are listed beneath **Values to repeat over**.
5. In **Values to repeat over**, select the check boxes for the categories you want to filter to.



For hierarchies, you must select **Hierarchy level to repeat over**.

Report pages for the selected analysis pages and visualizations will be created for one category at a time.

6. Combining filters might lead to empty results due to totally filtered out data. If you want to include also empty results in the report, for example for reference purpose, select **Include empty pages**.




7. Click **OK**.  
The dialog closes.
8. Click **Save**.

### Editing what is repeated over in a PDF export

A prepared report to be exported to PDF might, in the **Selected contents** pane, contain sections whose content should be repeated in the export. These sections are indicated by thin frames. You can edit what to be repeated over.

#### Procedure

1. Move the cursor over the upper right corner of a framed section, and click .  
Depending on what is repeated over, either the Repeat over bookmarks dialog opens, or the Repeat over filter values dialog.
2. Make your changes in the dialog according to the procedures in [Repeating pages and visualizations over bookmarks](#) on page 946 or [Repeating pages and visualizations over filter values](#) on page 947, and click **OK**.  
The dialog closes.
3. Click **Save**.

## Exporting to Microsoft PowerPoint

---

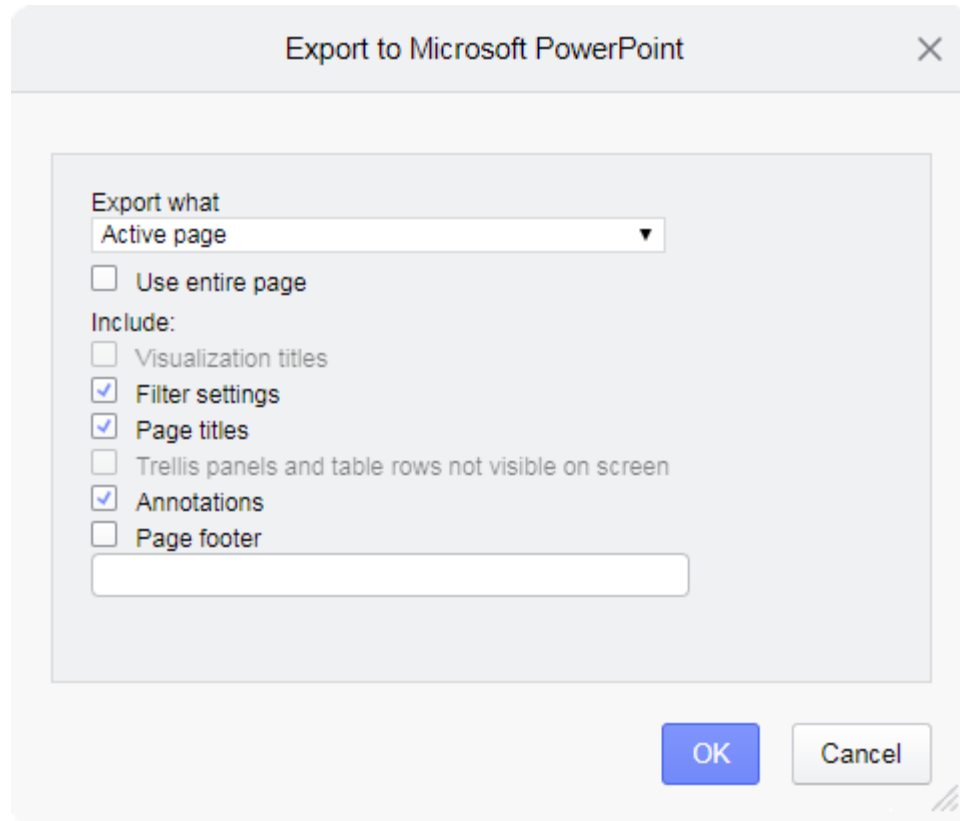
You can export an entire analysis or parts of it to a Microsoft PowerPoint document.

#### Prerequisites

The analysis to export is open.

## Procedure

1. On the menu bar, select **File > Export > To Microsoft PowerPoint**.



A dialog with available settings is displayed.

2. Specify the parts of the analysis to export, and then define the preferred export layout.
3. Click **OK**.

## Result

Depending on your web browser settings, the powerpoint file either downloads to your computer right away, or you are prompted to open or save it.

## Exporting to Microsoft Excel

You can export any tabular visualization, that is, the table, cross table, graphical table, or summary table visualizations, to an Excel file.

In general, the data is exported with the same data type and formatting as in Spotfire. However, in some cases, there is no direct mapping between the data type in Spotfire and the number format in Excel. Based on the data type, the data is then mapped to a default format, which is one of the predefined formats in Excel, for example General, Number, Date, or Time. This may cause the values to be displayed slightly differently depending on the culture set. To learn more about the predefined number formats in Excel, see [Microsoft's official documentation](#).

If it is not possible to map a data type exactly or the formatting differs significantly in the export, you will see a notification in Spotfire.

Binary data, except for images, cannot be exported. Also, zebra striping in table visualizations is not applied in the resulting Excel file.

## Prerequisites

You must have access to the Export Data license feature (under the Spotfire Enterprise Player license).

## Procedure

1. Click the visualization you want to export.
2. On the menu bar, select **File > Export > Visualization to Microsoft Excel**.



Alternatively, right-click the visualization, and select **Export > Visualization to Microsoft Excel**.



In the web client, you can select the option **Export table** to export all data in the visualization and to remove all layout and styling, such as font or colors, in the resulting .csv file. Select the option **Export table (without value formatting)** to also remove value formatting, such as Currency or Percentage.

3. If you are using the installed client, specify a file name and location for the Excel file, then click **Save**. If you are using the web client, the file either downloads to your computer, or you are prompted to open or save it, depending on your web browser settings.

## Exporting visualization to image

---

You can export any visualization as an image.

## Procedure

1. Right-click the visualization you want to export.
2. Select **Export > Visualization to image**.  
Alternatively, select **File > Export > Visualization to image**.
3. If you are using the installed client, specify a file name and location for the image file, then click **Save**. If you are using a web client, the file either downloads to your computer, or you are prompted to open or save it, depending on your web browser settings.



When using the installed client, you can also right-click a visualization and select **Copy Visualization Image** to copy the image to the clipboard.

## Exporting data

---

You can export data from your analysis either to a local file or to the library. When you export to the library, it is always data from a data table that will be exported, but when you export to a local file, you can choose between data from a data table or the data visible in a visualization.

### Exporting data to the library

The data that is exported to the library will be saved in an SBDF file (Spotfire Binary Data Format file), which stores the data and metadata from your analysis in binary form.

## Procedure

1. On the menu bar, select **File > Export > Data to library**.  
The Export data to library dialog is opened.
2. Select which **Data table** to export data from.  
Data that is located in an external data source (in-database) cannot be exported to the library, and will not be available for selection.

3. Select what data to export:
  - **All rows** - exports all rows available in the entire data table, regardless of filtering.
  - **Filtered rows** - exports the rows remaining after the current filtering (using the filtering scheme on the active page).
  - **Marked rows** - exports the rows marked in the visualizations (using the specified marking).  
Note that all marked rows will be exported, even if they are filtered out.
4. Click **OK**.
5. On the **Files and data** flyout, navigate to the library folder where you want to save the data, and specify a name.
6. Optionally, click **Edit description and keywords** to add additional information regarding the saved data.
7. Click **Save**.  
The data is saved as an SBDF file in the specified library folder, and can be reached by anyone with sufficient permissions to the folder.

## Exporting data from a visualization to file

You can export data from any visualization on the active page in the analysis, including the Details-on-Demand, if it is visible. When you export data from a visualization, you will get the aggregated values for the visualization items. For tables, cross tables, and summary tables, you can choose to export all data or only the marked data. For all other visualization types, you can export the data that is marked in the visualization. When exporting from the table visualization or the Details-on-Demand, a number of different file formats are available. For all other visualization types, the exported file will always be a tab-separated text file.

### Prerequisites

To export data from a visualization other than a table, cross table, or summary table, you must first mark the items you want to export.

### Procedure

1. On the menu bar, select **File > Export > Data to file**.  
The Export data to file dialog is opened.
2. Under **Export data from**, select **Visualization**.
3. Select the visualization to base the new data file on from the drop-down list.  
You can export all or marked data from tables, cross tables or, summary tables, or the marked data from any other visualization. You can also select to export the data currently shown in the Details-on-Demand.
4. If both options are available for the selected visualization, select what data to export:
  - **Data for all items** - Exports all data in the selected visualization, the way it looks right now. This means that only the data remaining after the current filtering will be exported. This option is available for tables, cross tables, summary tables, and the Details-on-Demand.
  - **Data for marked items** - Exports data for the currently marked items in the selected visualization. Note that all marked rows will be exported, even if they are filtered out. This option is only available if there are marked items in the analysis.

5. Select file format.

Data can be exported as a tab-separated text file (\*.txt) for marked items from any visualization.

For data from a table visualization or the Details-on-Demand, you can export the data to a text file, to an Excel file, to a comma-separated file (\*.csv), to a semicolon-separated file (\*.csv), to a Spotfire Text Data Format file (\*.stdf), or to a Spotfire Binary Data Format file (\*.sbdf). See below for more information on SBDF and STDF files. Note that when exporting to Excel, it is always the values that are exported and not the formatting, except for date/time values where the current locale is used.

6. Click **OK**.
7. If you are using the installed client, specify a file name and location for the file, then click **Save**. If you are using a web client, the file either downloads to your computer, or you are prompted to open or save it, depending on your web browser settings.

## Exporting data from the data table to file

You can export data based on one of the data tables in your analysis.

### Procedure

1. On the menu bar, select **File > Export > Data to file**.  
The Export data to file dialog is opened.
2. Under **Export data from**, select **Data table**.  
Note that this option is not available for external (in-db) data. Use **Export Data from > Visualization** when you want to export from in-db data tables.
3. In the drop-down list, select which data table to export data from.
4. Select what data to export:
  - **All rows** - exports all rows available in the entire selected data table, regardless of filtering.
  - **Filtered rows** - exports the rows remaining after the current filtering (using the filtering scheme on the active page).
  - **Marked rows** - exports the rows marked in the visualizations (using the specified marking).  
Note that all marked rows will be exported, even if they are filtered out.
5. Select file format.  
You can export the data to a tab-separated text file, to an Excel file, to a comma-separated file (\*.csv), to a semicolon-separated file (\*.csv), to a Spotfire Text Data Format file (\*.stdf), or to a Spotfire Binary Data Format file (\*.sbdf). See below for more information on SBDF and STDF files. Note that when exporting to Excel, it is always the values that are exported and not the formatting, except for date/time values where the current locale is used.
6. Click **OK**.
7. If you are using the installed client, specify a file name and location for the file, then click **Save**. If you are using a web client, the file either downloads to your computer, or you are prompted to open or save it, depending on your web browser settings.

**SBDF and STDF files**

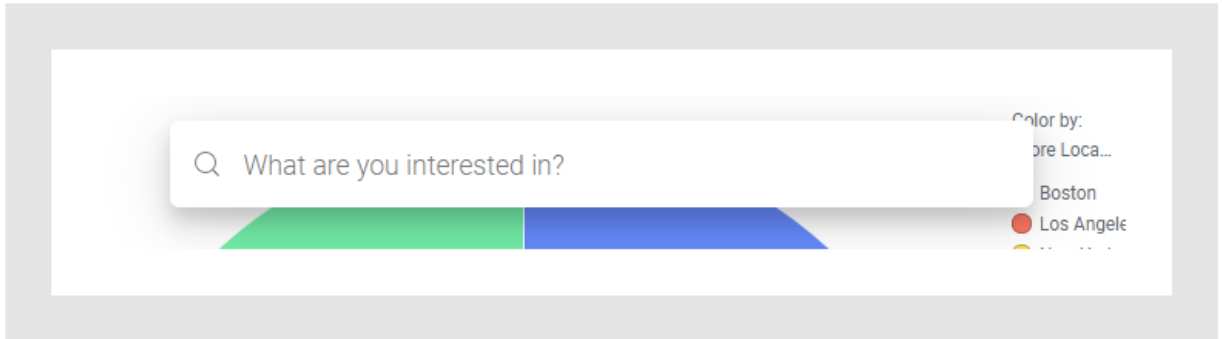
The Spotfire Binary Data Format (\*.sbdf) stores the data and metadata from your analysis in binary form. It can be used to greatly increase the performance when working with linked data in Spotfire.

The Spotfire Text Data Format also includes metadata. For example, the data types of the columns are saved, so you do not need to specify any import settings when opening files of this data format in Spotfire. If you select the Spotfire Text Data Format (\*.stdf) you will be able to open the file in older versions of Spotfire, but you might not get all functionality of the Spotfire Text Data Format.

# Find

---

The **Find** tool is a fast way to find contents in your data. It consists of a text field where you enter a search string and a list of results for the search.



To access the tool, click **Find**  on the menu bar, or **Ctrl+F**.

You can enter search criteria to find, for example,

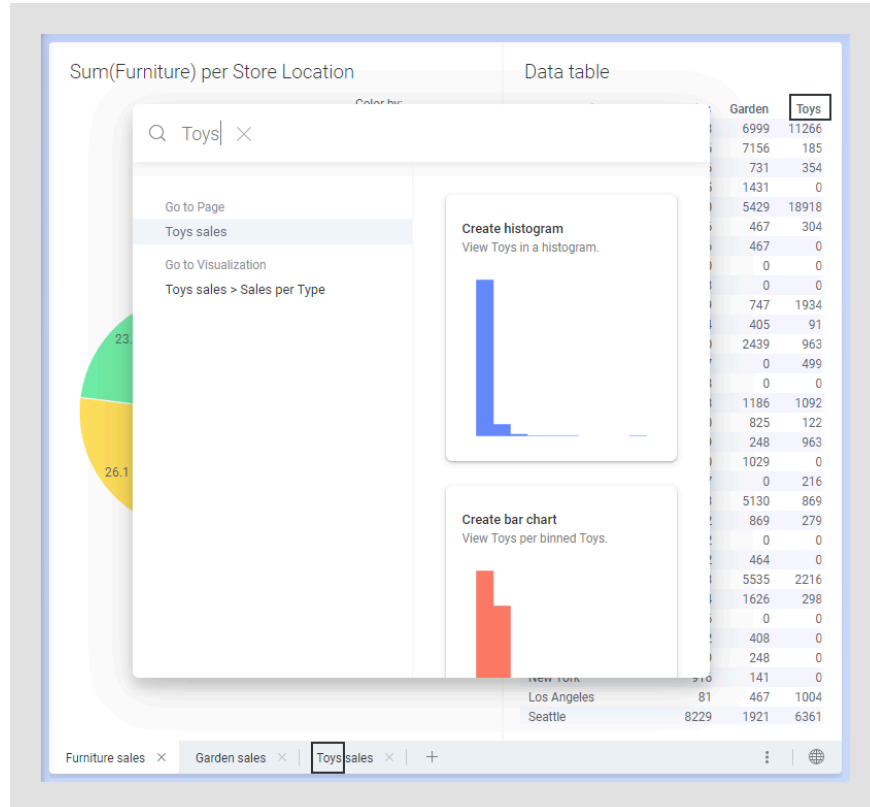
- actions that you want to perform
- pages and visualizations in your analysis
- certain data values for marking purposes
- library items that you want to open.

Last, but not least, you can

- enter data column names and get visualizations recommended to you.

## Example

In the analysis below, there are three pages (Furniture sales, Garden sales, and Toys sales), and a data column named Toys. If you enter Toys as search criterion, you get visualizations based on the Toys column recommended to you, which you can add to the analysis through a simple click.



Moreover, the search gives that a page includes Toys in its page title, and lists also the visualizations on that page. Simply click any of the listed items to access it.

Additional examples of what can be entered in the search field are:



- `y:sales` will list all visualizations with Sales on the y-axis.
- `sales category:page` will only list all pages with "sales" in the title.

## Syntax

You can also use logical expressions in your search strings. For information on the syntax for searches, see the [Searching in Spotfire clients](#) page.

## Searching in Spotfire clients

There are many places in the Spotfire clients where you can search for different items. For example, you can search for filters, analyses in the library or elements used to build information links in the



Information Designer. All of the available search fields use the same basic search syntax, which is presented below.

### Basic rules

- By default, a search will match against all words in the name, description or keywords. Searching for "Apple" will match every item that includes words that start with Apple, and will match items with the values "Apples are tasty" and "This is an apple".
- Search matches the beginning of all words. For example, "ple" will match Plenty and Plexiglas but not Apple.
- Search is not case-sensitive.
- Boolean AND is implicit when words in the search expression are separated by space. For example, "Apple Banana" will match anything where a word starts with Apple and another word starts with Banana.

Keyword	Example	Function
*	*ple *ple*	Finds items with a word ending in ple. Finds items where a word contains ple.
Quotation marks	"A Green Apple"	Finds items where the exact phrase A Green Apple is included.  (To find items starting with an exact phrase, you can add an asterisk inside the quote: " A Green Apple*".)
Double quotation marks	"9"" nails"	Finds items where the exact phrase 9" nails is included. Add a second double quote to escape a literal quote.
AND	Apple AND Fruit	Finds items with a word that starts with Apple and another word that starts with Fruit.
OR	Apple OR Banana	Finds items that include a word that starts with Apple or Banana.
NOT	Ban NOT *ana	Finds items that have a word that starts with Ban but does not end with ana. For example, Bangles and Banned would be found, but Banana would not.
( )	Apple and (Banana or Pear)	Used to group items in Boolean searches. See below for more information about searching for text within parentheses.
Quoted keywords	"and" "or" "not"	Finds strings that are protected keywords. Just typing and in the search field will not find anything since the word and is a protected keyword. If you need to search for the word " and", you must use quotation marks around it.
:	DataType:Integer	[Only applicable when searching for columns.]  The colon is used to search for attributes. In this example it finds columns where the column property DataType has a word that starts with Integer.  Almost any property can be used in the search, including custom properties. See <a href="#">Understanding properties in a Spotfire analysis</a> on page 273 for information about the available default column properties.

Keyword	Example	Function
::	Name::Apple DataType::Date	[Only applicable when searching for columns.]  Finds columns where the exact value of the property Name is Apple. In this example, a column named only Apple would match, but a column named Apple from Spain would not.  If you want to search for the Date data type, and not get any hits on DateTime columns, use :: instead of a single colon.
:<	DistinctValueCount:<10	[Only applicable when searching for columns.]  Finds columns with less than or equal to 10 unique values.
:>	DistinctValueCount:>10	[Only applicable when searching for columns.]  Finds columns with more than or equal to 10 unique values.
Null, Empty or not existing column property.	Tag:null or DistinctValueCount:null Keywords:null	[Only applicable when searching for columns.]  Finds all columns with no Tags or if the column property DistinctValueCount does not exist. Finds all columns with no keywords.

### Logic of search expressions

Search expressions are evaluated from left to right for logical operators with the same precedence. For example, the search expression:

```
Apple Banana or Pear
```

will be evaluated as

```
(( "Apple" ) AND ( ( "Banana" ) OR ( "Pear" ) ) )
```



If you cannot find what you are looking for, try adding more wildcards. For example, to locate a filter called "Sales (\$)", enter the search expression "Sales (\$\*", to avoid interpreting the text within the parenthesis as a Boolean expression. Adding quotation marks around the search string can also help in finding strings that include non-alphanumeric characters, in some situations.

# Saving an analysis

You can save new or updated analyses to the library.



If you are preparing analyses to be used by others you might want to review the section [Creating analyses for others](#) on page 962 before you save your analysis to the library.



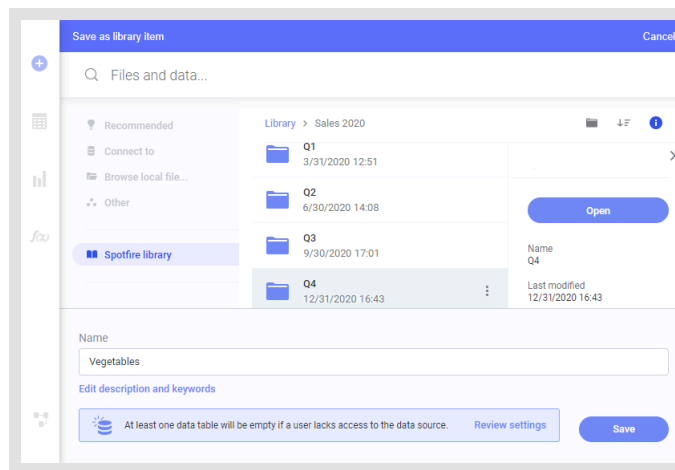
You might also want to [specify key columns](#) before saving the analysis. This is important if you want to reapply any specified tags when reopening the analysis file, or if you want to [reapply markings](#) in unaggregated visualizations (including markings in bookmarks).

If you want to save updates to the last saved version of the analysis, select **File > Save** on the menu bar.

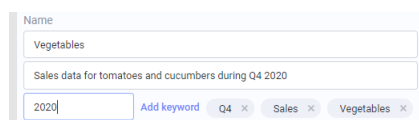
To save an analysis for the first time to the library, or with a different name, or at a new location, follow the procedure below.

## Procedure

1. On the menu bar, select **File > Save as > Library item**.  
The Files and data flyout is opened.



2. Navigate to the folder where you want to save the analysis file.
3. Specify a **Name** for the analysis file.  
The name can be 256 characters at most, and the following characters are not supported: \ / : \* ? " < > |
4. Optionally, click **Edit description and keywords** to show two additional input fields, where you can enter a description or add keywords specific to the contents of the analysis. Keywords are used to enhance organization and search capabilities. The image below shows a description and some keywords added to the analysis that will be saved.



5. Read the information in the blue section at the bottom of the flyout.  
It is a summary of what the implications will be to other users of the analysis if you save the analysis with the current data loading settings.
6. If you want to see a detailed view of the current data loading settings or edit them, click **Review settings**.

7. In the Data loading settings dialog, review the settings for all data tables in your analysis, and make changes if needed.

Depending on whether you want data to be updated from a linked source each time you open the analysis, or, whether you want to store a fixed version of the data in the analysis, you might want to change the data loading settings for a data table or a part of a data table. See [Load methods](#) for more information.



By properly configuring the data loading settings for all data sources in the [data canvas](#) in advance, the saving procedure becomes faster.

You can click the plus or minus sign for a data table to expand or collapse the data table and show or hide details about the various sources that were used to build the data table. Click on the drop-down list to change the data loading setting for a particular source.



If you want as much data as possible to be stored in the analysis, use the **Change all applicable data sources to** drop-down list and select **Stored data**.

8. Click **OK**.
9. Click **Save**.

### Result

The analysis file is saved to the library.



It is possible to save your analysis locally by selecting **File > Download as DXP file**.



An administrator can change the default data loading settings for file data sources, using preference settings. The preference settings are reached from the installed client, in the Administration Manager, **Preferences** tab, under **Application > DataImportPreferences**. The administrator can also change the preference default for embedded data or force embedding of data when saving to library, under **LibraryPreferences**. If the final data table is embedded, then you will not be able to change any data loading settings for specific sources.

## Deleting items from the library

You can remove items from the library in the **Files and data** flyout, or when browsing the library in a web browser.

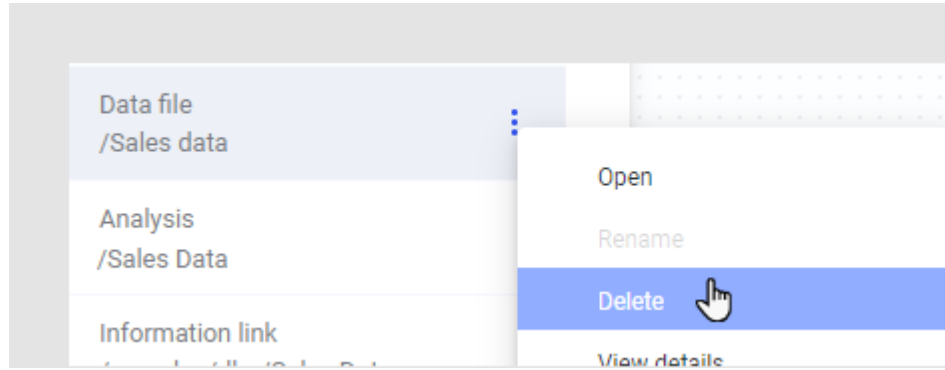
### Prerequisites

You have sufficient access [permissions](#) to modify the contents of the folder to delete items. To delete a folder, you must have **Modify** permissions for all subfolders in the underlying folder structure. To be able to delete, move, or edit an item from the library web view, you must also have the **Spotfire Business Author > Spotfire Business Author** license feature.

### Procedure

1. On the [authoring bar](#), click **Files and data** .
2. Locate the item you want to remove using search or by browsing the Spotfire library.

3. Click the menu next to the item and select **Delete**.



You can also select **File > View library**, select multiple items in the **Library** web view, and delete them all simultaneously.

## Reverting to a previous version of a file

If an analysis or other type of data file has become corrupted when it was modified, you can use the versioning functionality in the library to revert to a previous version of the file. You can also look at previous versions to review the library version history of a file.



Some components used within an analysis can be updated separately in the library (e.g., data functions, or visualization mods). If such a linked item is used in the analysis, you might not see the exact version that you saw earlier when reverting an analysis, unless you also revert the component.

Permissions to a file are always based on the latest version of the file and are never reverted.

### Prerequisites

You must have the right to add and edit files in the library to perform these steps.

### Procedure

1. Open the analysis you want to revert in a Spotfire client.
2. On the menu bar, select **File > View version history**.  
The library view of the selected file is opened in a web browser.
3. In the details field to the right, under **Version history**, you can view the different versions that have been saved in the library.
4. Right-click on the row of the version you wish to restore and select **Restore** (or use the menu at the end of the row).

### Result

The version you selected to restore will be added to the end of the list in the version history, and it will become the new current version. All previous versions are saved in the history.



You can also save a copy of a specific version as a new file by right-clicking it in the **Version history** list and selecting **Save version as new file**.



When working in the installed client, it is possible to switch off library versioning for a specific analysis file from the **Library** tab of the Document Properties.

## Adding library items as favorites

You can add any library item (a file, folder, data connection, and so on) as a favorite and have it show up in a specialized view in the library. This way, you can easily find your most important or most commonly used items in the library.

### Procedure

1. On the authoring bar, click **Files and data**.



You can also add items as favorites while using the web view of the library from **File > View library**.

2. Browse or [search](#) for the file, folder, or other item to add as a favorite.
3. On the row or card for the interesting item, or in the **Details** view, click the star icon, **Add to favorites**.

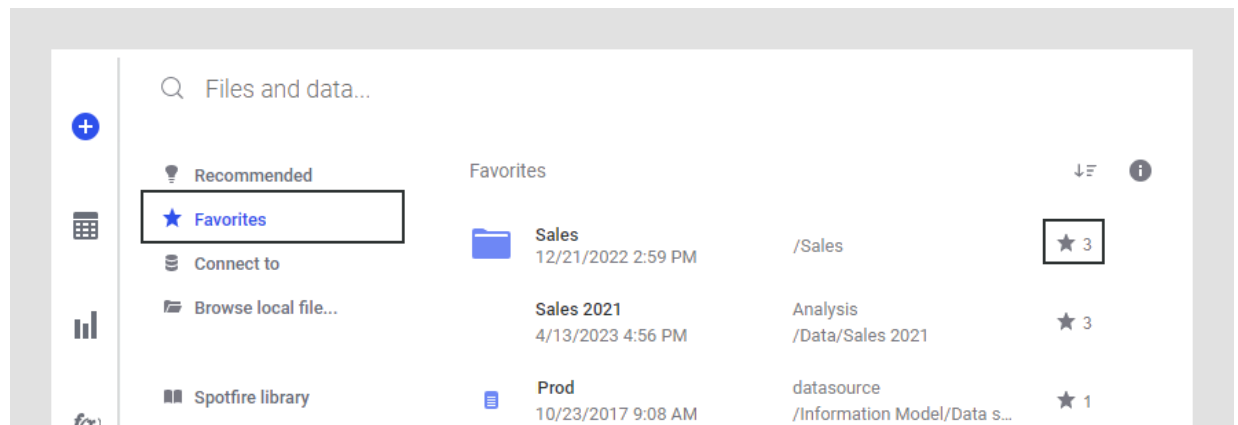


Click again to **Remove from favorites**.

### Result

The item is marked as a favorite and it will be shown in the **Favorites** view in the **Files and data** flyout.

If there is a number next to the star then this number shows how many unique users have added the item as a favorite and it can help you to locate popular or important analyses and data in the library.



An administrator can decide not to use this functionality in the library, as described in [Switching off the favorites functionality](#) in the *Spotfire® Server and Environment - Installation and Administration User Guide*.

## Creating analyses for others

When you are creating analyses to be used by others, there are certain things that can be good to think about. Depending on your audience and their licenses, access to data, or their work environment, you might need to think about your analysis design from different angles.

See the different sections for some specific tips and tricks. Many of the sections can be of interest for the same analysis.

### Dashboards

Dashboards are designed to present what is important in analyses, in an informative way. They can be used, for example, for analyzing, monitoring, or reporting purposes. Regardless if you design the dashboard to be consumed on a wall screen or a cellular phone, the data that is summarized in the

analysis must be presented in a structured layout. Key parts of the analysis should catch the consumer's eye instantly and the content should be understood without much effort from the user.

Dashboards might not only show a static view of the data; interaction capabilities can be built in for various purposes, such as drill-down into data details, and filtering data to certain values. The interactive dashboard can be designed so that the consumer gets instructions on the different tasks that can be performed, or the dashboard setup might even guide the consumer through the analysis in a specific order.

Not only the actual information is important; the look and feel of the dashboard also contributes to how a consumer perceives and understands the content. Using the most suitable visualization, a structured and well thought-out layout, legible texts, and colors that add value and are easy to distinguish enhance the understanding of the dashboard. For guidelines to make your dashboards work for more people, see [Designing accessible dashboards](#) on page 971.

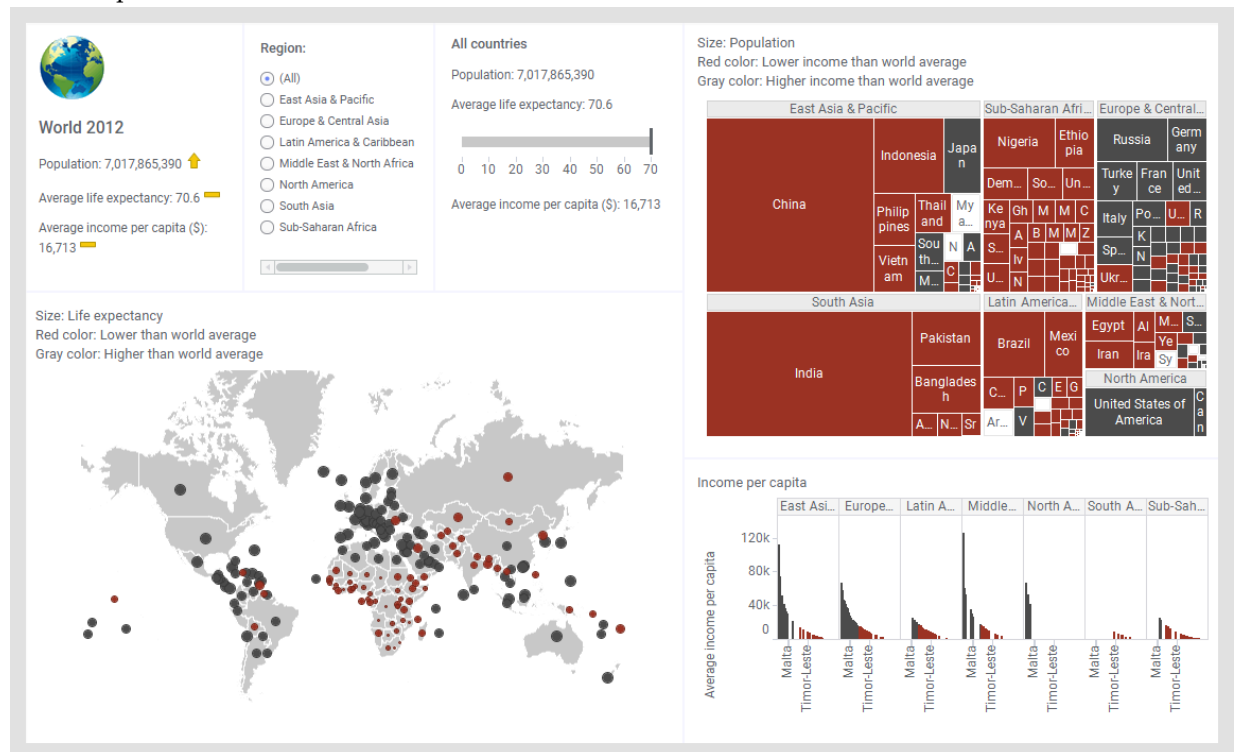
Many features are available to fulfill these different needs when creating dashboards. Most of the features are generally applicable, but are worth being gathered and accentuated in a dashboard context. See [Useful features when designing dashboards](#) on page 965.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

### Example of a dashboard

An example of an interactive dashboard is shown below.



It is based on the following data table, containing demographic data such as population, life expectancy, and income per capita for all the countries in the world\*.

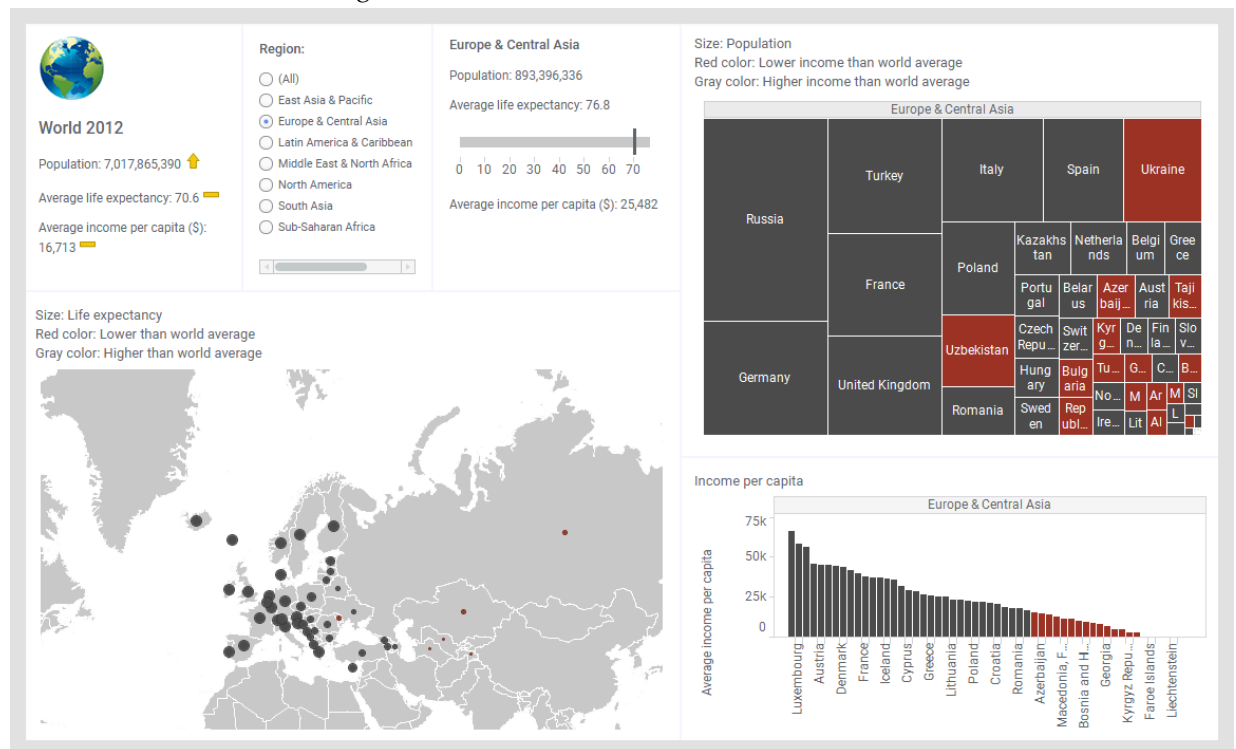
Country	Region	Population	Life Expectancy at Birth (years)	Income per capita
Bulgaria	Europe & Central Asia	7305888	74.31	15160
Bosnia and Herzegovina	Europe & Central Asia	3833916	76.12	9440
Belarus	Europe & Central Asia	9464000	72.06	16810
Switzerland	Europe & Central Asia	7996861	82.70	56120
Cyprus	Europe & Central Asia	1128994	79.64	28850
Czech Republic	Europe & Central Asia	10510785	78.08	26610
Germany	Europe & Central Asia	80425823	80.89	44310
Denmark	Europe & Central Asia	5591572	80.05	43890

In the upper left part of the dashboard, you are informed about the total population of the world, and the average life expectancy and average income per capita. As a supplement to the absolute figures, icons have been added to show the trend.

The visualizations are configured to provide overviews of relationships, real or potential ones, between some of the demographic variables:

- In the map chart, the colored markers differentiate countries with life expectancy below average from countries above average. Not only colors are used to visually provide information; the sizes of the markers also make it possible to compare the expected life lengths between countries.
- In the treemap, you get an overview of the populations in the countries (or regions) by comparing the sizes of the rectangles. Moreover, because the rectangle colors are defined to show whether the income per capita is higher or lower than the average in the world, you might find relationships between population sizes and incomes per capita.
- In the bar chart, you can compare the incomes per capita across the different regions, and across the different countries within each region.

This dashboard also offers interaction possibilities. If you select a Region radio button, you can look into details for the selected region.



Above, Europe & Central Asia is the selected region, and the visualizations adapt to show only data for this particular region. The details for the region are found in the text area to the left of the treemap.



The text at the top of the text area shows what region is selected. Moreover, to visually compare the life expectancy in the region to the world average, a bullet graph is used.

Note, however, that the numbers regarding the entire world have been configured to stay unaffected.

\* The data is attributed to The World Bank. The data is provided for demonstration of Spotfire functionality. Disclaimer of Endorsement or Liability - The World Bank does not endorse or recommend the use of Spotfire in any way. The World Bank and Cloud Software Group, Inc. make no warranties for this demonstration data.

## Useful features when designing dashboards

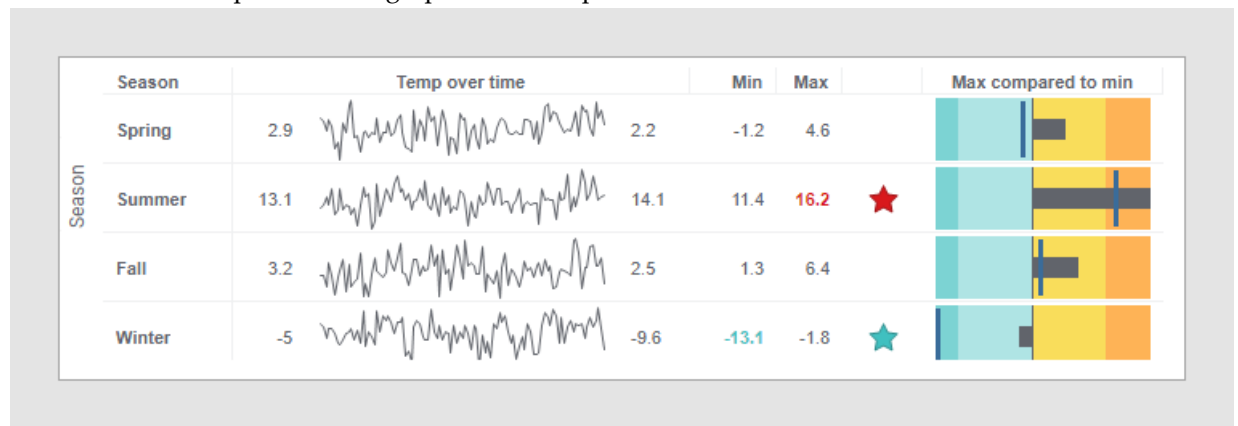
Numerous features are available to make dashboards work and look the way you want. Most of them are generally applicable when working with analyses; they are not specific to dashboards. The features that are listed below are, however, described from a dashboard perspective. Some of them affect the dashboard as a whole, and some are only tips on minor settings that might be useful. The features are not listed in any specific order.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

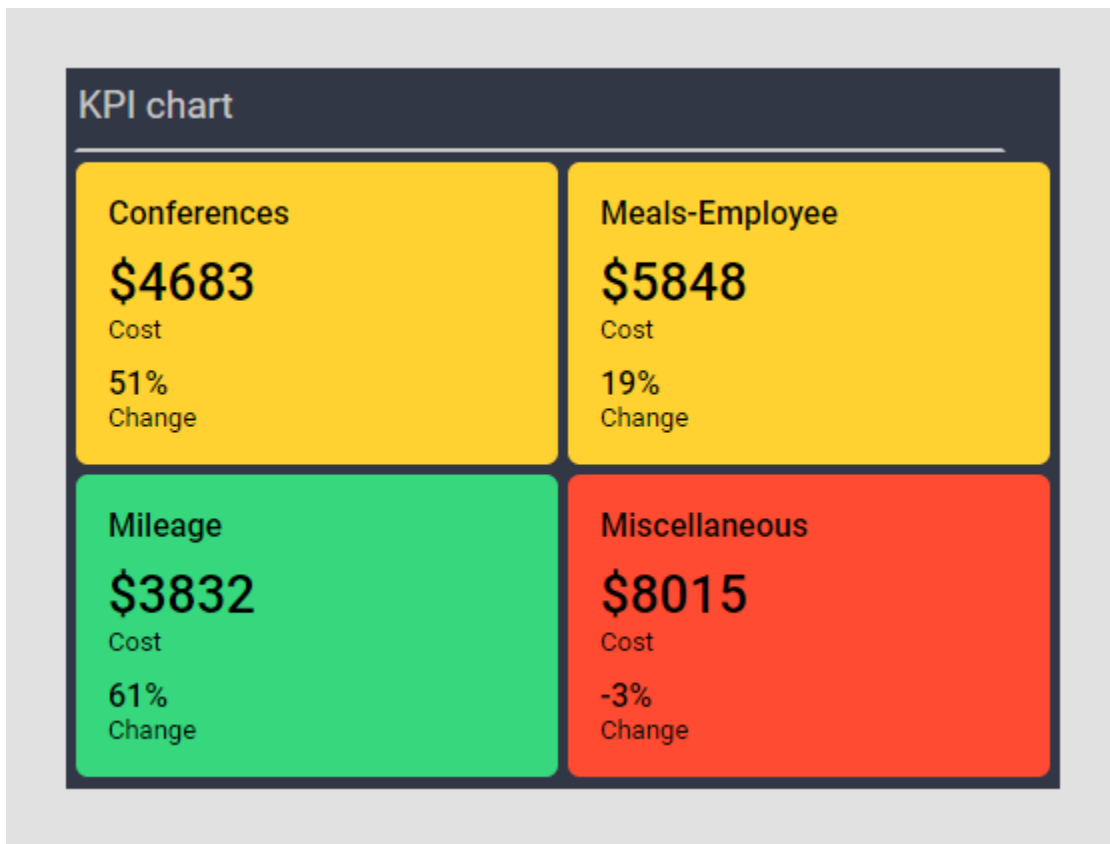
### Quick and compact visual summaries

It might be important to quickly get an overview in the dashboard, and the more detailed information can be left out for the moment. For example, you might need to immediately spot specific values, see trends, make comparisons, or get noticed about issues. For these purposes, you can use a [graphical table](#) (authored in the installed client), which presents information in a condensed format through display of sparklines, calculated values, icons, and bullet graphs that update dynamically. The image below shows examples of these graphical table options.



Sparklines, calculated values, icons, and bullet graphs can also be shown in text areas (authored in the installed client).

Another visualization that provides information at a quick glance is the [KPI chart](#) (added in the installed client). It shows KPIs, Key Performance Indicators, which measure factors that are crucial when monitoring the current performance of, for example, a company. By letting colors indicate the performance, you can easily scan the current situation. The KPI chart is well suited for small screens.



### Navigating in the dashboard

A dashboard can consist of more than one analysis page, and there are different ways to navigate the user through them. You might want to use page tabs with descriptive titles, which the user can choose from, or you might want to guide the user through a step-by-step workflow, that is, in a certain order by clicking Next-, Previous-, and numeric links. Another option to control the workflow is the use of clickable items in text areas, graphical tables, and KPI charts. Then it might be beneficial to switch off other navigation modes.

To determine the most suitable option for your dashboard, see more details in [Changing page navigation method in an analysis](#) on page 605 and [Guided analyses](#) on page 973.

### Descriptive naming of items

If the names of items are too long, or difficult to understand, you can change them. You can rename the following:

- [data tables](#)
- [data columns](#)
- [pages](#)
- visualizations (by changing the title in the visualization properties)
- [axes \(column selectors\)](#) (by setting a display name or using the "AS" keyword)

### Text areas

The use of [text areas](#) (authored in the installed client) on the analysis pages is a powerful means to add the following:

- Explanatory and instructional texts.
- Images for decoration or information purposes.
- Links, buttons, and images, which, when clicked, perform actions such as open web sites, switch to another page, apply a bookmark, execute a script, or refresh a calculation.
- Controls from which you handle the filtering in the analysis. See also *Filtering schemes* below.
- Controls from which you handle various configuration properties such as switching column on an axis.

You can use the text area features, for example, to control how people should navigate through the analysis, and to gather controls in the same place instead of using different panels, which might free space.

## Filtering schemes

It might be useful to add [filtering schemes](#), which give you more freedom to control how the filtering should work in the dashboard. By default, any filtering is applied to the entire analysis, that is, to all pages. The use of filtering schemes makes it possible to apply different filtering to the various analysis pages, and even to the separate visualizations on a page.

If you use filter controls in a text area, you open up a capability that is not available when using filters in the **Data in analysis** flyout or the **Filters** panel; if visualizations on the same page use different filtering schemes, you can filter one visualization in one way and another visualization in another way. This is possible because, in the text area, each filter control is connected to a selected filtering scheme.



You can also let a visualization stay unaffected by any filtering by switching off the filtering completely, under **Data limiting** in the visualization properties.

## Hiding redundant information

When designing a dashboard, it is worth considering whether some information might be redundant and therefore could be removed. Unnecessary information clutters the dashboard and distracts from what should be the focus. Besides, removing information frees up space. Tips on what you can do are:

- Adjust (or maybe even remove) the [legend](#) to show only relevant items. Consider hiding the title and column selector of items in the legend.
- Hide panels that are not used. Keep in mind that the text area can be used instead for certain controls, for example, specification of filters and properties.
- If the **Filters** panel must be visible, [hide filters](#) that are not used.
- Consider whether the (All) and (None) options should be visible in radio button filters.
- Decide whether the visualization title bar adds value or should [be hidden](#). Sometimes any needed information can be shown in visualization descriptions instead.
- Consider whether [selectors](#) and [scale labels](#) should be visible.
- In the map chart, [hide controls](#) that are not used.
- In the table visualizations, [hide columns](#) that are not of interest.
- When action controls are used for guiding the user through the analysis workflow, consider whether page navigation at the bottom or top of the window must be available.

## Adapting to different screen sizes

Dashboards might be consumed on different screen sizes, from cell phones to wall screens. There are a lot of features that are useful when designing dashboards for the different platforms. For information, see the following topics:

- [Layout options](#)
- [Maximizing the canvas](#)
- [Authoring analyses for small screens](#)
- [Modifying responsive layouts](#)
- [Specifying minimum acceptable page size](#)
- [Locking the size of certain visualizations](#)
- [Specify a Canvas size in the Document Properties](#) (installed client only)
- [Preparing analyses for Spotfire web clients](#)

What is most important is testing that the created dashboard works on the devices where it is intended to be used.

## Interaction

Dashboards might be designed for viewing only, or for interactivity. If the consumer is supposed to perform certain actions, make sure there are instructions available in text areas (see the Text areas section above). For example, give instructions on which filters to use or properties to change, or which items to mark to view details in a details visualization. As mentioned earlier, you can place links and buttons in text areas, which, when clicked, execute specified actions. An example of an action is applying a bookmark.

## Look and feel

There are plenty of visual appearance features that can be used to design a good-looking dashboard.

- The canvas [visual theme](#) impacts the overall look and feel. You can easily switch between the predefined visual themes, light and dark, alternatively customize your own visual theme. In the Edit custom theme dialog, you can change the general appearance of borders, background colors, gaps, and many other details.
- In the visualizations, predefined [color schemes](#) are available for selection. You can import other color schemes, or create your own.

To make the dashboard easier to grasp, use consistent colors. For example, if a green bar in a bar chart represents the category 'Male', use the same color also elsewhere for the 'Male' category.

Consider using colors to draw attention to, or indicate, what is important.

Make sure the colors used in the visualizations are distinguishable. See also [Designing accessible dashboards](#) on page 971.

- You can add images in the text areas, for example, logotypes.

## Dashboard features examples

The intention of the following examples is to give hints of what can be achieved, and inspire you to further investigations of the Spotfire features to get the dashboards you want.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

## Switching off filtering in a visualization

A typical interaction in a dashboard is filtering the data. By default, any filtering you do affects all visualizations on the page, if they are based on the same data table. There might be times, though, when you want a visualization to stay unaffected of the filtering, for example, if you want to use it as a reference. Then you can switch off the filtering for that particular visualization.

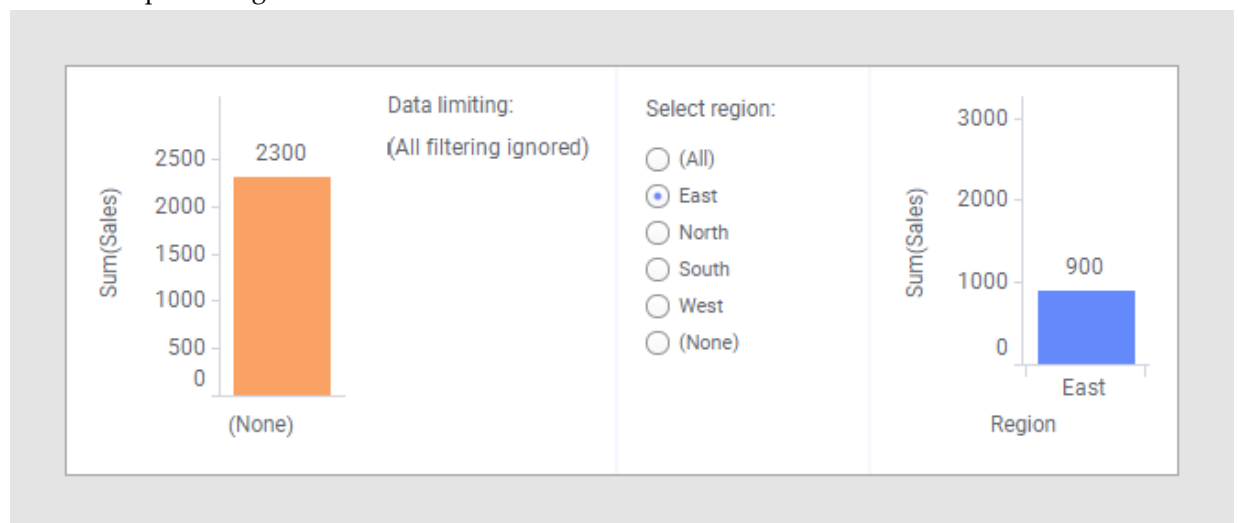
Go to the **Properties** for the visualization, and locate the **Data** section. Under **Data limiting**, remove all filtering limitations.

### Example

The bar charts below are based on the following simple data table.

Region	Sales	Month
South	200	March
West	150	March
North	250	March
East	400	March
South	50	April
West	300	April
North	450	April
East	500	April

In the bar chart to the left, you can see the total sales for all regions, and in the bar chart to the right, you see the sales for the region that is selected in the filter. This way, you can view the entire sales and the sales for a specific region at the same time.

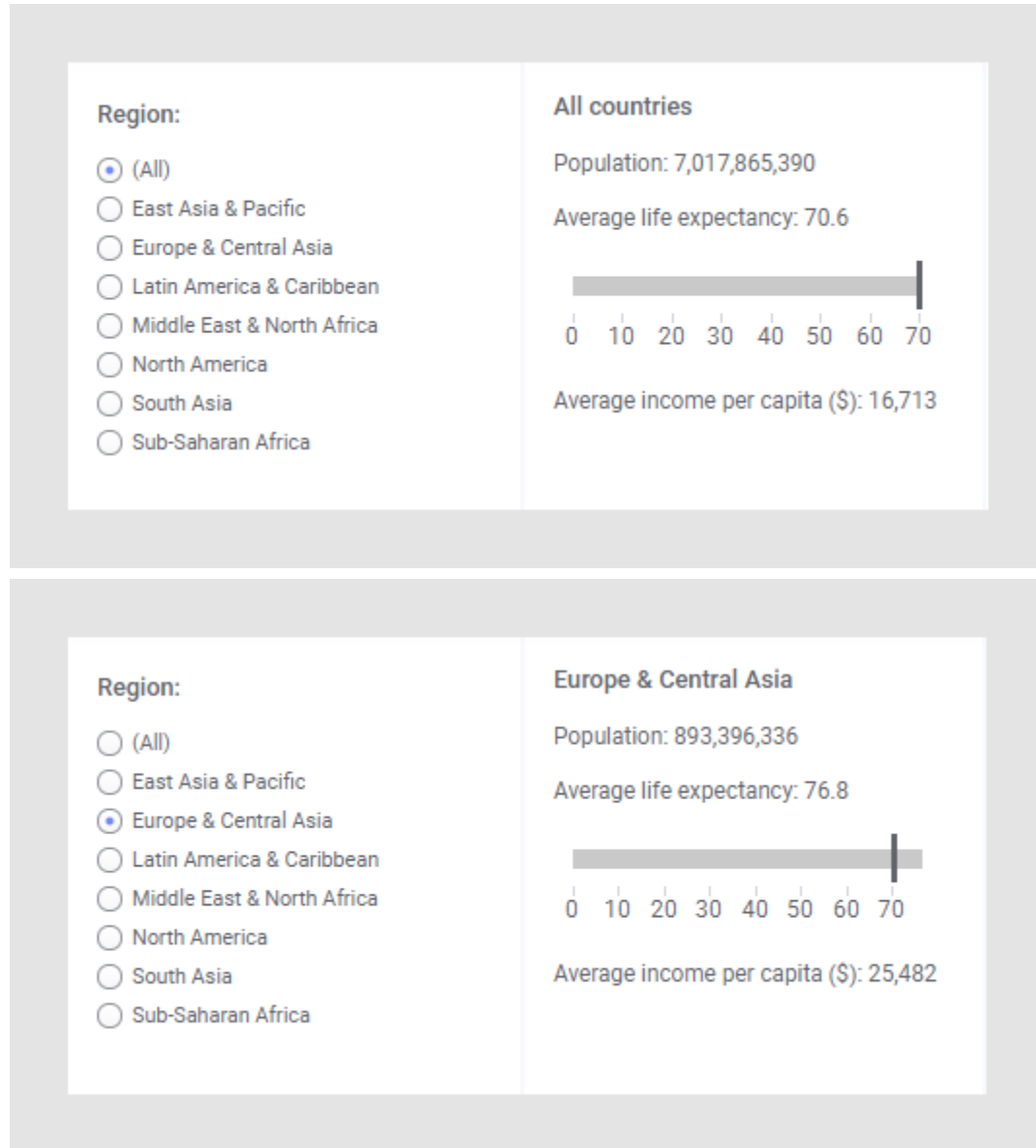


This is possible because the filtering is switched off for the left bar chart by removing **Use the current filtering from the page** (and any other filtering) from the visualization properties, under **Data**.

Another alternative to compare all data to filtered data is the use of [Subsets](#).

## Making text dynamic

The images below are excerpts from the example in the [Dashboards](#) topic. Note that not only the numbers are changed when interacting with the dashboard; the top text to the right also reflects what is currently selected in the Region filter.



How this is done is described below.

In the left text area, a filter control has been added that lets you select a region, and then demographic data for that region is displayed in the text area to the right. To update the text to show the region that is currently selected, a dynamic item is used.



Text areas are authored using the installed client.

1. Right-click the text area and select **Edit Text Area**.
2. Click **Insert Dynamic Item**, and select **Calculated Value**.
3. In the **Settings** dialog for the calculated value, select **Values**.
4. Right-click the **Calculate values using** selector, and choose **Custom Expression**.

5. In the Custom Expression dialog, enter the expression:

```
case UniqueCount([Region])
  when 0 then "None"
  when 1 then First([Region])
  else "All countries"
end
```

6. Close the custom expression and calculated value settings dialogs.
7. In the **Edit Text Area** dialog, click **Save**.

The dynamic Calculated value item shows the evaluation of the expression. If a region is selected in the filter control, the name of that region is displayed, and if the (All) option is selected, the result is 'All countries', respectively.

## Designing accessible dashboards

When creating dashboards for a large audience you should be aware of how you can make your visual analytics dashboards more accessible.

Here are a number of guidelines to help you design and construct accessible dashboards.

### Use colors with high contrast and be consistent

1. Provide contrast for items in visualizations, at least 3:1.  
For colors in your visualizations and items that people will interact with, pick colors that have a contrast ratio of at least 3:1 against the background color.
2. Provide contrast for text, at least 4.5:1.  
For colors of text elements in your dashboard, pick colors that have a contrast ratio of at least 4.5:1 against the background color.
3. Pick colors that are color-blind-friendly.  
For colored items that will be compared, pick colors that are color blind friendly. Use color-blind-friendly palettes and avoid using red and green together.
4. Visualizations of the same type should be consistent in a dashboard.  
If there are two or more visualizations of the same type in a dashboard (for example, two pie charts) these should be consistent. This means that they should have similar settings as well as the same "look and feel". However, they do not necessarily have to use the same colors. For example, keep the legends on the same side for visualizations of the same type.
5. In addition to color, use other visual means to provide information.  
Avoid relying on color only to differentiate items in visualizations. Consider adding differences in size, shape and patterns, so that the items can be distinguished from one-another.

### Make your content descriptive and readable

1. Provide descriptive visualization titles.  
The title should describe the content in a visualization and introduce the consumer to the visualization.
2. Provide descriptive legends.  
Legends in visualizations must describe the values clearly.

3. Provide descriptive page titles.

Page titles should describe the content that the consumer can expect on the page.

4. Provide text alternative for visualizations.

There should be a text alternative present for all visualizations to aid those that cannot see or for some other reason cannot comprehend the visualization. This should include a description of the visualization and its purpose, as well as potential generated insights.

5. Provide an overview of the dashboard.

Include a description of your entire analysis. What is the purpose of it, how should it be navigated and what should the consumer expect?

6. Use sans-serif fonts.

Pick a sans-serif font such as Roboto or Arial. Avoid fonts such as Times New Roman because it is more difficult to read for some people.

7. Provide text larger than 16 pixels.

If possible, use text that is at least 16 pixels tall. If smaller text is used it requires higher contrast than step 2 under "Use colors..." above suggests.

### **Minimize the complexity of your visualizations, and the dashboard as a whole**

1. Provide a logical structure in lists and tables.

Tables and lists should be structured so that it is easy to navigate them. Use logical structures, such as alphabetical order, or north-south for geographical data.

2. Minimize visual and navigational complexity in dashboards.

Strive for a clear and simple dashboard to minimize complexity. If visualizations are meant to be viewed together, put them on the same page and in the order they are meant to be viewed.

3. Dashboards should be constructed with a natural flow.

Design your dashboards so that they are easy to perceive and navigate sequentially. There should be a natural flow, for example, from top-bottom and left-right. Place important information at the top of the dashboard.

## **Reports**

One strength of Spotfire is to present what is important in the data by means of interactive dashboards. However, sometimes there is a need to present the information in another format, for example, as standardized PDF reports or as Microsoft® PowerPoint® slide decks. To publish in different formats, the features under **File > Export** are used.

### **PDF reports**

Standardized non-interactive reports of the findings in the analyses can be prepared in advance and then exported to PDF files. Many options are available for customization of the reports. You can prepare reports that export different views of the analysis. For example, visualizations and analysis pages can be repeated for the different values in a data column. This way you can mirror certain analysis interactivity, even if the report is 'static'.

To exemplify, say that you have sales data for some stores of different types in different regions of a country. In the report, it might be interesting to first visualize total sales per store type for the entire country, and then let the visualization be filtered and repeated for one sales region at a time.



If the analysis contains tables or trellised visualizations that are not fully visible on an analysis page, it is still possible to let the report contain the entire table or visualization.

If you want printouts of the reports, you simply print the PDF files.

These [prepared reports](#) can work as templates, that is, they can be reused and exported over and over again, to show the state of the most recent data.

Using Spotfire® Automation Services, the creation and distribution of new PDF reports can be automated and scheduled to take place at certain intervals. It is possible to, automatically, open an analysis from the library, create and export a prepared report to a PDF document, and then even e-mail this PDF document to all concerned. For example, one use case could be that e-mails containing the sales figures for the previous week are sent every Monday morning to all managers in a company.

For more information, see the [Spotfire Automation Services User Guide](#).

## PowerPoint reports

An entire analysis, or selected parts of it, can be [exported to PowerPoint](#) slides.

## Guided analyses

You might want to prepare the analysis file to load particular data and show certain visualizations, but you might also want to instruct other people how to use the document. There might be a specific order a person should go through the pages, and you can add detailed instructions on what to look for and which filters are relevant to use on each page. To aid in this, there are a number of things you can do in Spotfire to define a guided flow through your document.

When the recipients of your analysis file open it, they are guided through the analysis as per your instructions, but they can do their own filtering and look closer at any noteworthy aspects they find interesting. This allows you to configure a generic analysis covering a subject such as sales over the entire United States, but you can instruct the recipients to filter down to the state they work in.

Some methods you can use to make your analysis guided are:

- Create a cover page.
- Write instructions in [text areas](#) (authored using the installed client), or in descriptions for each visualization.
- Place links or buttons leading to relevant tools, pages or views in the text areas.
- Switch the [page navigation](#) to step-by-step mode, or define your own page navigation using links and the history arrows navigation mode.
- Use customized [filtering schemes](#).
- Keep in mind the intended end users' level of data access.

### Create a cover page

Create a cover page for your document and explain the purpose of the analysis on this page. Tell the recipients what kind of data is included in the analysis, and the possible results to look for. When you save the analysis file, before sharing it with your colleagues, make sure the cover page is active so that the analysis file will open showing that page first.

### Write instructions in text areas

You might want several pages in your document. For example, the first page might show a map chart of the United States covering the overall sales results. The second page might have a bar chart comparing

the sales and expenses in different states. The third page might show sales figures for each individual salesperson.

It is often good if each page includes some text where you explain what the page shows and its purpose. Give the reader some instructions on which filters are relevant to modify, and perhaps mention that they can mark interesting items in the visualization and see more information about those items in the Details-on-Demand window or in a details visualization. For example, on the second page with the bar chart comparing states' sales and expenses - you could ask the recipients to select the radio button that filters down to the state they work in.



You can include filters directly in a text area, to help users finding the functionality when they need it. See [How to Use the Text Area](#) in the *Spotfire Analyst User Guide* for more information.

The use of [property controls](#) in the text area can also be a great help when creating analyses for other people. Just remember to add instructions regarding any constrictions for the control and inform about the purpose of the control using regular text.

### Place links or buttons to relevant tools or views in the text areas

In some cases, it might be relevant for the recipients to perform an action; to filter out values, to go to another page, to apply a bookmark or to open a panel from the menu bar, and so on. Instead of writing a long instruction, you can create a link or a button in the text area that performs this action when clicked. The instruction you do write in the text area can be simpler, like "When you have marked the items of interest, click [here](#) to filter to filter out all other items from the visualization." Links and buttons are a very powerful way to allow even casual users of Spotfire to perform analysis of data in a fast and easy manner.

Links or buttons can also include bookmarks that show a specific view of the data, and thereby helps explaining the steps that have led to a particular conclusion. Using custom properties to define the visualizations, and property controls that allow the end users to easily change the property values can further simplify the analysis procedure for many people.

### Use step-by-step or navigate using action controls

When you want to emphasize that the recipients of your analysis file should step through the pages in a certain order, you should change the page navigation from titled tabs to [step-by-step navigation](#). This means that the pages will instead be shown as numeric links, together with a Previous and Next link, in the page navigation area. The recipient of the analysis file will then start on the cover page, if you have selected to show it, or on the first real page. By clicking Next, you will step through each page in order, performing the analysis described in the text areas along the way.

This can be very powerful because, by default, the filtering done on one page affects all other pages as well. You can therefore create a flow where the first page allows the recipients to filter out unwanted data by looking at one visualization. Then, they can proceed to the next page where they continue to drill down into the data, filtering out more unwanted rows using another visualization, and so on.

Using [action controls in text areas](#) and in graphical tables and KPI charts (authored using the installed client), you can define your own navigation flow. This way, clicking an action item directly on the pages is the way to move forward within the analysis. When you use this type of navigation, it might be beneficial to select **Page Navigation > Off** (that completely switches off other navigation modes) or **Page Navigation > History Arrows** (that allows return to previously visited pages).

### Use customized filtering schemes

By default, the filter settings are the same for every page in the document, and they will stay the same until you start changing the filtering schemes. The filtering schemes give you complete control and complete freedom to decide which pages of your guided analysis should affect one another and which

should not. You can keep the same filtering scheme for all pages, create a different one for each page, or assign the same filtering scheme to two or more pages.

Applying different filtering schemes can be useful if your guided analysis consists of several separate analyses originating from the same data table. When two pages use the same filtering scheme, the filtering performed on one page is carried over to the second page and vice versa. If they use different filtering schemes, the filtering on each page only affects that very page.

For example, you can create one page where State is the only activated filter, permitting the recipients to click through and compare sales for the states without being bothered with any other filters. (If you want to, you can also use [organize filters](#) (in the installed client) to hide all unused filters.) On the next page, you can apply a different filtering scheme relevant to another analysis of the sales data, and so on.

### Keep in mind the intended end users' level of data access

Always make sure the end users of your guided analysis have access to the same data sources as you do. [Permissions](#) to analyses and information links are handled using the Library Administration tool (in the installed client), or directly from the Spotfire Server Library web page, in an on-premises environment, but you might also need to consider access to databases when using data connections. See also [Preparing analyses for Spotfire web clients](#). You can also consider adding prompt steps that could limit the available data for each end user when you are adding the data.

## Preparing analyses for Spotfire web clients

When an analysis file is saved to the library, it becomes available to use in Spotfire web clients. Web browser-based versions of Spotfire can be used for viewing and exploring prepared analyses, but also, in some cases, for creating and editing analyses. Any analysis can be saved to the library and opened with a Spotfire web client, but there are a few things to keep in mind that can make things easier for the web client users. First, consider how familiar your target audience is with the visualized data. The familiarity of the target audience can affect the instructions required as well as the configuration of the analysis. For example, you might need to change the visibility of some filters for a certain audience. Second, bear in mind that all users of the analysis might not have access to the same data sources.

To share an analysis you have created using the installed client with a web client user, you can copy the link to the analysis in the library from the confirmation dialog that is shown after you have saved the analysis. Paste the copied link, in an email or similar, to share the information with your colleagues. They can also use search in the library to find the analysis.

### Tips when preparing analyses for Spotfire web clients

1. Use text areas or descriptions in visualizations to write instructions about the analysis and how it is configured. For instance, if a details visualization is used, try to explain the relationship to the main visualization that controls it.
2. Use informative titles for the visualizations.
3. The overview of data will be improved if you hide filters that are not relevant for the visualizations in the analysis. For an explanation of how to hide filters, see [Showing and Hiding Filters](#) in the *Spotfire Analyst User Guide*.
4. When using links or buttons in a text area, try to write instructions so that the information is helpful, even if the links should happen to become temporarily unavailable.
5. Consider whether all users of the analysis have access to all of the original data sources. If not, you might need to save some data tables, or parts of data tables, [stored](#) in the analysis, rather than linked to the original data sources. See [Permissions](#) in the *Spotfire Analyst User Guide* for information about how permissions are handled in the library. There might also be user authentication limitations directly on the data source itself. Talk to your Spotfire Administrator or data manager to understand

how the permissions for different user groups have been defined within your company. Sharing analyses in a Cloud environment is different than in an on-premises environment.

6. Before using scripts in an analysis, consider the restrictions due to security reasons. For more information, see [Usage of Scripts](#) in the *Spotfire Analyst User Guide*.
7. It is not possible to view the 3D Scatter Plot on the web. Also note that some types of custom visualizations might not be shown. The analysis can still be opened in web clients, but visualizations that are not supported will not be shown.
8. Consider whether web client users should be allowed to export data from tables. In the installed client, you can change these settings on the Appearance page of the Table Properties, Cross Table Properties, Summary Table Properties, and/or Details-on-Demand Properties.
9. Consider whether web client users should be able to open a personalized view of the analysis and/or be allowed to add bookmarks. In the installed client, you can open the [Document Properties dialog](#) to change these settings.
10. Consider whether web client users with sufficient permissions should be allowed to edit the analysis. In the installed client, you can open the [Document Properties dialog](#) to change these settings.

### Design for the intended platform

If you know that your end users will view an analysis using a particular equipment or with a specific browser, you can look up the current visualization area size on that device and then design your analysis so that it is optimized for that particular screen size. Note that the current visualization area size is dependent on what browser and which toolbars and items are shown in the browser, so make sure that you are using the same settings when creating the analysis.

For more information about designing for a small screen, see [Authoring Analyses for Small Screens](#). For more information about adding properties specific to mobile use, see [Bar code scanning](#) and [Location](#).

If you want to create an analysis that works both on a computer with a large screen and also on a projector, you can use bookmarks to switch between the different sizes and layouts. The sizes for the projector and the large screen computer can be stored by an administrator in the [preferences](#).

### Using bookmarks for adapting an analysis to different screen sizes

1. Configure an analysis that works well on your large screen with the intended fixed size preset. Then, configure all text areas to **Include configuration in bookmark**.
2. When the document is finished, add a bookmark named something like "Large screen computer".
3. Switch to the fixed size preset for the projector, and make updates to make the analysis work on a projector:
  - Change the size of text areas and fonts.
  - Change font sizes for titles, axis labels, and so on, using **Tools > Options, Fonts** page (installed client only) and **Apply to Document**.
  - Update the layout, remove unnecessary legends, and so on.
4. When the projector version of the analysis is finished, add a bookmark named "Projector".
5. Save the analysis.

The analysis is now adapted both to a projector and to a large screen desktop computer.



If you are preparing an analysis containing streaming data, WebSockets must be enabled on the Spotfire Server running the web client. You can open the analysis in a web client and verify that the tooltip on the globe icon reads 'Connection: WebSockets'. If it does not, WebSockets might not be enabled. Contact your Spotfire administrator for more information.

## Adding functionality for mobile devices

When designing analyses to be used on mobile devices, you can add functionality to take advantage of this. For example, you can include bar code scanning, or functionality that reads the devices current location.

### Bar code scanning

You can add a document property to an analysis designed to be used on an iOS mobile device, to get the ability to scan machine readable codes (such as QR codes or bar codes). The reserved document property name is `AppMachineReadableCode` and it is of the data type `String`.

An analysis that contains this document property shows an active bar code icon on an iOS mobile device. If the document property is not included in the analysis, the bar code icon is not available. The value of the document property is set by scanning a supported code using the mobile device. It is set to the empty string when no supported code is scanned.

The following machine readable code types are supported:

- UPC-E
- Code 39
- Code 39 mod 43
- EAN-13 (including UPC-A)
- EAN-8
- Code 93
- Code 128
- PDF417
- QR
- Aztec

You can use this functionality to scan a code, and, by creating a table visualization limited by an expression, filter the data to the item with the scanned code. For example, this can be used for inventory control. See [Adding a bar code reading capability](#) on page 977 and [Creating an expression to limit the data to the bar-coded entry](#) on page 979 for detailed instructions.

For more information about designing for iOS mobile devices, see [Designing Spotfire® Analytics for Small-Screen Display](#).

### Adding a bar code reading capability

By adding the document property `AppMachineReadableCode` to your analysis, you can activate the bar code reader capability of an iOS mobile device that uses the analysis.

#### Prerequisites

Some of the functionality described here can only be authored or accessed using the installed Spotfire client.



The data table you import must have a column containing the data for either a bar code or a QR code. This column must be of the data type `String`.



When you import the data using the installed client, you can check the Import Settings dialog to make sure the column containing the code has the right data type. You can also [change the data type](#) later.

### ***Adding bar code reading capabilities for your mobile users***

#### **Procedure**

1. Open data that contains a string column of bar codes or QR codes.
2. Create a **Table** visualization.
3. On the menu bar, select **File > Document properties**, and in the Document Properties dialog, click the **Properties** tab.
4. Click **New**, and in the dialog, provide the reserved name `AppMachineReadableCode`.
5. Set the **Data type** to String. Save the changes.  
You can leave the default value blank.

#### **Result**


The bar code is added.

Your visualization now has the document property required to activate the bar code reading icon on an iOS device (available for a consumer user, in the cloud or on premises).

Next, add a text area to show the code that the iOS mobile device scans.

### ***Adding a text area to show a scanned code***

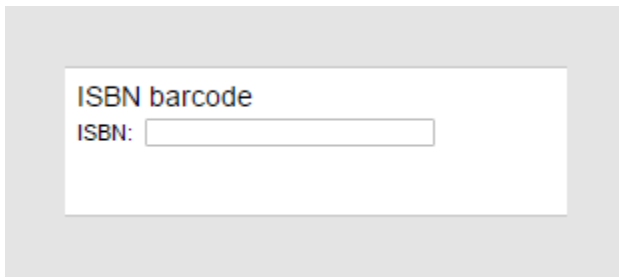
#### **Procedure**

1. On the authoring bar, click  to open the **Visualization types** flyout.
2. Drag the **Text area** to the wanted position on the analysis page.  
A blank text area is created.
3. Right-click the new text area, and click **Properties**.
4. In the dialog, provide a title, such as "Scanned code", and then close the dialog to save the change.
5. Click the **Edit text area** icon, and in the Edit Text Area dialog, click the **Insert Dynamic Item** icon and choose to add a **Calculated Value**.
6. Under **Values**, right-click the selector under Calculate values using and select **Custom Expression**.
7. In the **Custom expression** dialog, add the expression `${AppMachineReadableCode}` and click **OK**.  
This expression inserts the document property value as text. See also [Properties in expressions](#) on page 907.
8. If desired, in the Edit Text Area dialog, provide a text label for the dynamic item.
9. Resize the text area to contain just the added item and the title.

#### **Result**

The text area can now display a scanned code.

You now have a text area where a scanned bar code or QR code will be shown when a code is scanned by an iOS mobile device using the analysis. For example, the following image shows a text area for containing the ISBN code, which appears as a bar code on the backs of books. The text box will display a captured ISBN code when the code is scanned using a mobile device.



Next, [create an expression to filter a visualization to the scanned bar code](#).

### Creating an expression to limit the data to the bar-coded entry

When bar code reading capabilities have been added to the analysis, you can add a limiting expression on a visualization to filter to only the last scanned bar code.

#### Prerequisites

Before you perform this task, you must have added the document property `AppMachineReadableCode` to the analysis, as described in [Adding a bar code reading capability](#) on page 977.

Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

In the next step, limit the data to show only the row that matches the scanned item.

#### Procedure

1. On the menu bar, click **Data > Add calculated column**.
2. Add the following expression:
 

```
If('${AppMachineReadableCode}' = '' or [barcode] = '${AppMachineReadableCode}',
    true, false)
```

 where `[barcode]` is replaced by the string column containing your bar code data. In the examples we have used a column called "ISBN" with ISBN codes.
3. Provide a name, such as "IsBarCodeFiltered" and check to make sure the expression is of type Boolean.
4. Save and close the dialog.
5. Create or locate the visualization to limit and open the **Visualization properties**.
6. Go to the **Data** section.
7. Under **Data limiting**, click **Add limitation** and select **Expression**.
8. In the **Limit Data Using Expression** dialog, select the column `isBarCodeFiltered`, and then click **Insert Columns**.
9. Click **OK** to close and save the change.

#### Result

When the value of `IsBarCodeFiltered` is set to `False`, all data is shown. When the data is scanned using the bar code scanner on an iOS device, `IsBarCodeFiltered` is set to `True` and the data is limited to show the entry corresponding to the bar code only.



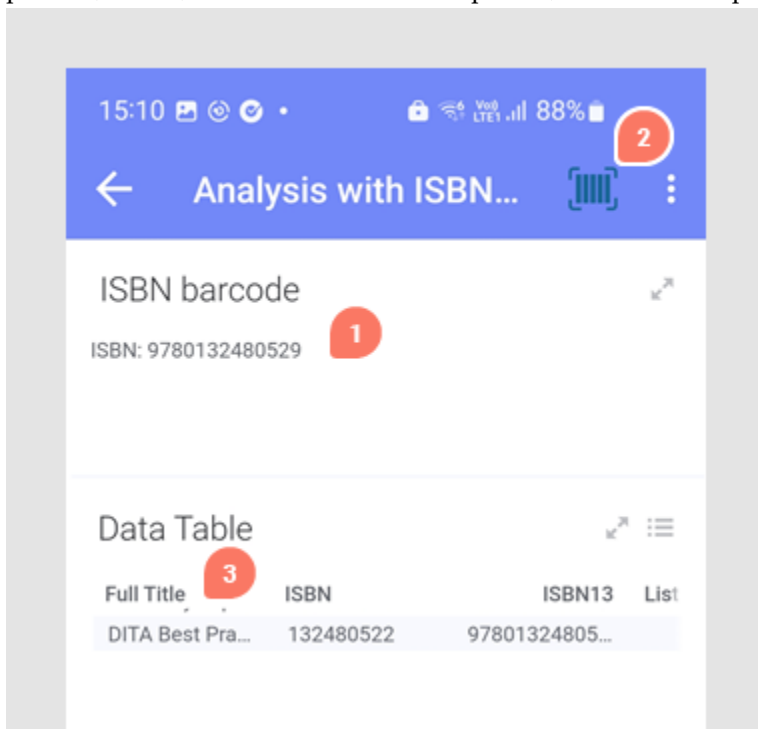
## Testing bar code filtering

### Procedure

1. From your iOS mobile device, open the analysis containing bar code scanning capability.
2. Tap the bar code icon.
3. Scan the bar code (or QR code) of an item that is listed in the data table.

### Result

The following image shows the text area and the table visualization after the bar code button has been pressed, that is, the value of the code captured, and the corresponding data in the filtered table.



1. Filtered ISBN entry.
2. The Bar code capture button has been used.
3. The title associated with the ISBN entry in the filtered table.



This analysis is for demonstration purposes and is not ideal for viewing on a small screen. For more information and advice on optimizing for a small-screen display, see the document [Designing Spotfire Analytics for Small-Screen Display](#).

### Location

Spotfire reserves two document properties for adding location services (or geo-positioning) to an analysis. Such an analysis is designed to be used on an iOS mobile device. The document properties are named AppGeoLatitude and AppGeoLongitude and they are both of the data type Real.

An analysis that contains these document properties shows an active location icon on an iOS mobile device, such as an iPad or iPhone. If the document properties are not included in the analysis, the positioning icon is not available.



When you add the document properties to a Spotfire analysis, set them both to 0.0. (The mobile app interprets these values as setting the location to inactive.) When the iOS device user taps the location icon in the app, the current latitude and longitude are passed to the document properties.



A data table with columns containing latitude and longitude can take advantage of the Spotfire expression method [GreatCircleDistance\(\)](#) to create a calculated column containing the distance between the device location and all locations in the data. Multiply the resulting value by the radius of the planet.

- For kilometers use 6371.
- For miles use 3959.

For example, the following expression gives the distance in kilometers:

```
6371*GreatCircleDistance(${AppGeoLatitude}, ${AppGeoLongitude},
[latitude_column_name], [longitude-column_name])
```

For detailed instructions and examples, see [Adding location capabilities to an analysis](#) on page 981 and [Finding your location on a map using the Spotfire app](#) on page 984.

For more information about designing for a small screen, see [Designing Spotfire® Analytics for Small-Screen Display](#).

## Adding location capabilities to an analysis

If your data has latitude and longitude columns, you can create an analysis that an iOS mobile device user can access to determine the distance from a location. Just add the reserved document properties and a calculated column for distance calculation, and then share your analysis with mobile users. The document properties activate the location icon in the mobile device. The following example demonstrates how to create such an analysis and test it.

The following examples demonstrate how to create such an analysis and test it.

### Adding the location properties

The first step to use location capabilities is to add the reserved properties to the analysis.

#### Prerequisites


Document properties must be added in the installed client.

#### Procedure

1. In the installed client, open a file containing the data with latitude and longitude columns.



In the installed client, you can use the **Import Settings** dialog to make sure the column names and data types are correct. You can always change column names and data types at a later time from the expanded **Data in analysis** flyout (in all clients).

2. On the authoring bar, click  to open the **Visualization types** flyout.
3. Drag the **Map chart** to the wanted position on the analysis page.
4. On the menu bar, select **File > Document properties**, and in the **Document Properties** dialog, click the **Properties** tab.
5. Click **New**, and in the resulting dialog, provide the reserved name AppGeoLatitude.
6. Set the **Data type** to **Real**, and the **Value** to 0.0 (the equator).

For testing purposes, you can set AppGeoLatitude to your latitude, but before saving the final version, set it to 0.0.

7. Add another document property, providing the reserved name `AppGeoLongitude`, setting the **Data type** to `Real` and setting the **Value** to 0.0 (the prime meridian).

For testing purposes, you can set `AppGeoLongitude` to your longitude, but before saving the final version, set it to 0.0.

### Result

Your analysis now has the document properties required to activate the location icon on a mobile device that can open it (available for a consumer user, either on TIBCO Cloud Spotfire or on premises).

## Adding distance calculation capabilities for mobile users

When the location properties is in place, you can use them to calculate distances between the current location and one or marked locations in a data table.

### Procedure

1. On the menu bar, select **Data > Add calculated column**.
2. In the **Expression** field, enter the following expression:

```
6371000*GreatCircleDistance(${AppGeoLatitude}, ${AppGeoLongitude}, [Latitude], [Longitude])
```



This calculation shows distances in meters. To specify miles, use 3959. To specify kilometers, use 6371.

The example uses *Latitude* and *Longitude* to reflect column names. If yours differ, modify the example.

3. Give the new column a name, such as 'Distance', and click **OK** to save and close the dialog.

### Result


Your analysis now includes a column that you can use to measure distances between the iOS device and any data point in the data table by using the geo-positioning properties.

## Adding visualizations for app users to see location and distances

### Prerequisites

Text areas must be authored in the installed client.

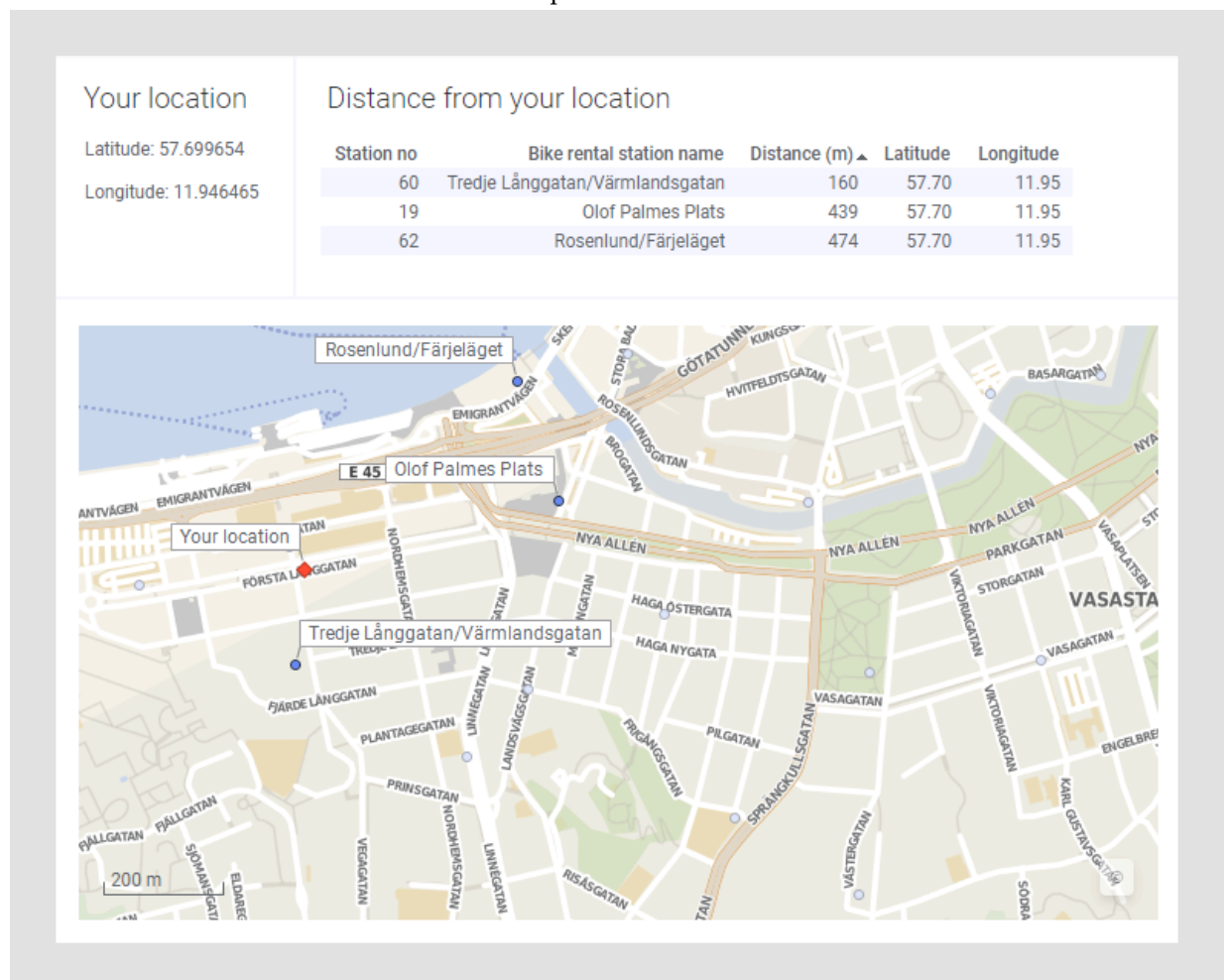
### Procedure

1. In the installed client, open the analysis containing the location data and the reserved document properties.
2. On the authoring bar, click  to open the **Visualization types** flyout.
3. Drag the **Map chart** to the wanted position on the analysis page.
4. Open the flyout again and drag the **Text area** to the wanted position on the page.
5. Right-click the new text area, and then click **Properties**.
6. In the dialog, provide a title, such as 'Your location'. Close the dialog to save the change.
7. Click the **Edit** icon.
8. In the Edit Text Area dialog, type "Latitude:" and click **Insert Property Control > Label**.
9. In the Property Control dialog, click to select the document property `AppGeoLatitude` and click **OK**.

10. In the Edit Text Area dialog, type "Longitude:" and click **Insert Property Control > Label**.
11. In the Property Control dialog, click to select the document property AppGeoLongitude and click **OK**.
12. Click **Save** and then close the edit mode of the text area.
13. Right-click the map visualization, and from the menu, select **Create details visualization > Table**.
14. Right-click the table visualization, and then click **Properties**.
15. Provide a name for the visualization, such as 'Distance from your location'. You might also want to adjust the order of the columns in the table so the new 'Distance' column is visible. Close the dialog to save the changes.
16. Test the visualization by selecting a data point or a range of data points on the map.  
The details visualization shows the selected data rows, with the calculated column reflecting the distance from the latitude and longitude shown in 'Your location'.
17. Save the analysis and share with your mobile users.

## Result

This example shows bike rental stations in a Swedish city. Each rental station has a location specified by latitude and longitude. You can see the distance (measured in meters) from your current location to the three rental stations that are marked in the map chart.



## Finding your location on a map using the Spotfire app

You can add a marker showing the current latitude and longitude in a map visualization, identifying where iOS mobile app users are currently located when they tap the location icon in the app.

### Prerequisites

Before you complete these tasks, you might want to work through the tasks in [Adding location capabilities to an analysis](#) on page 981.

## Adding expressions to find your location

To show a marker with your current location, you must first create a new data table, containing three calculated columns based on the location properties.

### Procedure

1. Create a single-column data table that contains a name row and one placeholder data row using Notepad or Excel.



You can create a CSV file containing a column name (for example 'Location') and add a second row with a placeholder string (for example 'My location') in the file. Save it with any name; in this example we use 'Location'.

2. Open Spotfire and add the placeholder data table.  
You can add it to an already existing analysis, such as the one described in [Adding location capabilities to an analysis](#) on page 981.
3. If they do not already exist, add two document properties: AppGeoLatitude and AppGeoLongitude, as described in the section *Adding the location properties* in [Adding location capabilities to an analysis](#) on page 981.
4. From the menu bar, select **Data > Add calculated column**.
5. In the resulting dialog, make sure you have selected the new 'Location' data table, add the expression `${AppGeoLatitude}`, and in the **Column name** field, provide a name such as 'lat'. Click **OK** to save the change.
6. Create another calculated column in the 'Location' data table by adding the expression `${AppGeoLongitude}`, and providing another name, such as 'long'. Save the change.
7. Create another calculated column in the 'Location' data table, using the following **Expression**:  
`Not(${AppGeoLatitude}=0.0 and ${AppGeoLongitude}=0.0)`  
The data type of this expression should be Boolean by default, with the value of `True` (unless you have used a mobile app to change the current latitude and longitude to something other than 0.0/0.0). This expression determines whether the two document properties (AppGeoLatitude and AppGeoLongitude) are set to 0 and 0, which is the intersection of the equator and the prime meridian. If they are not set to 0 and 0, then a marker is added to the map at the point where the latitude and longitude values intersect. (If the values are 0 and 0, then no marker is shown.)
8. In the **Column name** field, give the new column a name such as 'isLocationSet'.

### Result

You now have a data table with three calculated columns that you can use to mark your current location in a map chart visualization designed for the mobile Spotfire app.

## ***Adding a marker to show your current location***

When you have created the data table containing your current location as described above, you can add this data table as a marker layer in a map chart visualization.

### **Prerequisites**

These steps must be authored in the installed client.

### **Procedure**

1. Create a new map chart, or use the one you created in [Adding location capabilities to an analysis](#) on page 981.
2. Right-click the map chart, and select **Properties**.
3. In the map chart properties, select **Layers**.
4. Click **Add > Marker Layer**, and add the 'Location' data table.
5. In the marker layer settings for Location, select **Colors**, and change the color for **All values** to a color that stands out. For example, if your data points are all blue, you could make the location color red.
6. Optionally, click **Shape**, and select a shape to distinguish the marker, such as an X or a star.
7. Locate the **Data** section of the Marker Layer Settings, and, under **Data limiting**, add a limitation based on an **Expression** (also known as **Limit data using expression**).
8. In the **Limit Data Using Expression** dialog, select the column `isLocationSet`, and then click **Insert Columns**. Click **Close** to close and save the change.  
The column is added as an expression. When the value of `isLocationSet` is set to `False`, the marker in the location map layer is hidden.

### **Result**

In the map, if the value of latitude and longitude both are not 0, and if the `isLocationSet` is `True`, the marker of the specified shape and color is shown on the map at the current latitude and longitude.

This example shows bike rental stations in the Swedish city Gothenburg. The text area at the top shows the current location as Latitude and Longitude. The column `isLocationSet` is set to `True`, which means that the Location marker layer displays a red marker at the current latitude and longitude.

Q Data in analysis...

Location

NUMBERS

- lat
- long

CATEGORIES

- ☒ isLocationSet
- Location

isLocationSet

☒ True

Your location

Latitude: 57.699654  
Longitude: 11.946465

Distance from your location

Station no	Bike rental station name	Distance (m)	Latitude	Longitude
18	Rosenlundsplatsen	847	57.70	11.96

200 m

29 of 29 rows 1 marked 10 columns Rental stations

Clear selections

# Sharing analyses

---

You can share your insights with others from the library, from an analysis, and from inside a visualization.

## Share an entire analysis

If you want your colleague to open an analysis you have created and saved in the library, you can [copy a link to the analysis](#) and then share the entire analysis without any modifications by sending the link in an email. Your colleague can then click on the link to log in and access your analysis in the library. If your colleague has access to the installed client, you can also [download the analysis as a DXP file](#) and then send the file to your colleague instead.

## Share selected parts of an analysis

If you would rather choose a few points of interest in an analysis to share with others you can do so. By [creating a PDF](#) with some interesting visualizations or pages, you can share your insights with others without having to consider whether they can log into Spotfire or not.

# Copying a link to an analysis

---

You can copy the link to a specific analysis to easily share it with others.

## Copying the link to the analysis that you are currently working on (web client only)

In the web client, if you want to get the link for sharing the analysis that you are currently working on in Spotfire, click **File** on the menu bar, and in the drop-down list, select **Share > Copy link**. Then, in the [Copy link dialog](#), copy the link shown in the **Link to analysis** field.

If desired, you can modify what to include in the link using the [Advanced](#) options.

## Copying links to analyses from the Files and data flyout

In any client, when browsing the **Files and data** flyout, you can open the details menu (**Click for options**) on the row of the interesting analysis, select **Copy link to item**, and choose the type of link that you want to use.

## Copying the link to an analysis from the Library app

You can also get the link to share an analysis file while you are browsing your Spotfire library. You can do this in the Library page on the Spotfire Server.

### Procedure

1. To go to the full library view, select **File > View library** on the menu bar.
  - a) You might be prompted to sign in with your credentials to open the Library app.
2. In the Library view, browse the library to find the analysis that you want to share.
3. Right-click the analysis file, and in the drop-down list, select **Copy link**.  
You cannot copy links to information links, SBDF files, or folders.
4. In the **Copy link** dialog, copy the link shown in the **Link to analysis** field.
5. Click **Close**.

## Result

You can now paste the copied link into an email and send the email to a colleague with whom you want to share the analysis.



You can also obtain links pointing to specific [bookmark](#) views, links containing a configuration block, and so on. See [Links to analyses in the library](#) on page 989 for an overview of different link types and how to retrieve them.

## Copy link dialog

In the web client, when sharing an analysis via **Share > Copy link**, you can specify exactly what to include in the shared analysis.

The options in the expanded **Copy link** dialog are listed below.

Option	Description
<b>Link to analysis</b>	The URL to the analysis. Copy the URL to share the analysis with others.
<b>Advanced</b>	Click <b>Advanced</b> to specify which of the listed menu options and UI elements to include in the shared analysis.
<b>Bookmark</b>	If you want to share a specific bookmark in the analysis, specify that bookmark in the drop-down list.
<b>Include the following menu options</b>	Note that to be able to use any of the optional menu options, the Toolbar also has to be enabled in the shared analysis.
Close	Select this to show the <b>Close</b> menu option in the shared analysis.
Download as DXP file	Select this to show the <b>Download as DXP file</b> menu option in the shared analysis.
Analysis information	Select this to show the <b>Analysis information</b> menu option in the shared analysis.
Filters panel	Select this to show the <b>Filters</b> menu option in the shared analysis. Note also that if the Filters panel is displayed in the original analysis, clearing this checkbox will also hide the Filters panel in the shared analysis.
Details-on-demand	Select this to show the <b>Details-on-Demand</b> menu option in the shared analysis. Note also that if Details-on-Demand is displayed in the original analysis, clearing this checkbox will also hide the Details-on-Demand in the shared analysis.
Collaboration	Select this to show the <b>Collaboration</b> menu option in the shared analysis. Clearing this checkbox will also hide any existing comments in the shared analysis.
Help	Select this to show the help menu options in the shared analysis. Includes the menu options <b>Help topics</b> , <b>Spotfire Support</b> , and <b>Spotfire Community</b> .
About	Select this to show the <b>About Spotfire</b> menu option in the shared analysis.
Log out	Select this to show the <b>Log out</b> menu option in the shared analysis.
<b>Include the following UI elements</b>	



Option	Description
Header	Select this to show the web client customizable header in the shared analysis.
Toolbar	Select this to show the toolbar in the shared analysis.
Toolbar menus	Select this to show the menus on the toolbar of the shared analysis.
Undo/Redo	Select this to show the <b>Undo/Redo</b> buttons on the toolbar of the shared analysis.
Viewing/Editing	Select this to show the <b>Viewing/Editing</b> drop-down list for users with sufficient permissions in the shared analysis.
Login button	Select this to show the <b>LOG IN</b> button on the toolbar if the user is not logged in when accessing the shared analysis.
Page navigation	Select this to allow page navigation in the shared analysis.
Status bar	Select this to show the status bar in the shared analysis.

## Links to analyses in the library

When an analysis has been saved to the library, you can reach it or share it using a number of different link types. A bookmark URL can be included to guide other people to an interesting aspect or starting point in an analysis.



Some of the functionality described here can only be authored or accessed using the installed Spotfire client.

See the table below for an overview of the link types available.

Link type	Description	Retrieve where?	Use when or where?
Link for users of the installed client	<p>Opens the analysis in your default version of an installed Spotfire client, for example Spotfire Analyst.</p> <p>Examples:</p> <pre>spotfire:server:https%3A//myspotfireserver/:analysis:f73c4ce4-9174-48bb-a602-bfff3c6d862e</pre> <p>or</p> <pre>tibcospotfire:server:https%3A//myspotfireserver/:analysis:f73c4ce4-9174-48bb-a602-bfff3c6d862e</pre>	<p>In any client, by right-clicking an analysis in the Spotfire library section of the <b>Files and data</b> flyout, and selecting <b>Copy link to item</b> &gt; <b>For installed clients</b>.</p> <p>In the installed client, by right-clicking an analysis in the Library Administrator and selecting a link to the library identifier for the installed client.</p> <p>In the installed client, in the <a href="#">Document Properties</a> dialog.</p>	Anywhere where you know that the recipients of the link have access to an installed Spotfire client and you want the analysis to be opened in a new instance of Spotfire.

Link type	Description	Retrieve where?	Use when or where?
-"- (with a bookmark)	<p>Can include a bookmark, if copied from the <b>Bookmarks</b> popover.</p> <p>Example:</p> <pre>tibcospotfire: server:http\://  myspotfireserver/: analysis:/Data/  MyAnalysis: bookmark:7c0669f2- 9701-44d4-96b3- 7fef9e129fa</pre>	In the installed client, from the <b>Bookmarks</b> popover, click the arrow to the right on the row of a bookmark and select <b>Copy Bookmark URL &gt; Link for Users of the Installed Client</b> .	-"-
-"- (pointing to server but not analysis)	<p>Point an installed Spotfire client to a certain Spotfire Server.</p> <p>Example:</p> <pre>tibcospotfire: server:http\://  myspotfireserver/</pre>	Manually edited.	You can also use this type of link to point the end users to a specific server without determining which analysis to open.
-"- (with a configuration block)	<p>Opens the analysis with a configuration block.</p> <p>Example:</p> <pre>tibcospotfire: server:https%5C://  myspotfireserver/: analysis:/Data/  My%20Analysis: configurationBlock: SetFilter  (columnName='\ Month\ ', values={\ February\ '});</pre>	<p>Manually edited.</p> <p>When pointing to a path to an analysis, the path must be URL encoded. In the installed client, you can copy the path with encoding from the <b>Library Administration</b> tool. The configuration block can then be added to the path.</p>	See <a href="#">Create a configuration block</a> on the <a href="#">Community</a> for information about configuration blocks.
Link that lets the recipients choose client	<p>Opens a web page where you can select whether to open the analysis in a web client, an installed client, or to download the analysis.</p> <p>Example:</p> <pre>https:// myspotfireserver/  spotfire/redirect? analysis=f73c4ce4- 9174-48bb-a602- bfff3c6d862e</pre>	<p>In any client, by right-clicking an analysis in the Spotfire library section of the <b>Files and data</b> flyout.</p> <p>In the installed client, by right-clicking an analysis in the <b>Library Administrator</b>.</p>	<p>When you do not know whether the recipients of the link want to open the analysis in a web client, in an installed client, or to download the analysis.</p> <p>For example, when sharing a URL in a blog post or sending an email with the URL to people from many departments that have access to different Spotfire solutions.</p>

Link type	Description	Retrieve where?	Use when or where?
-"- (with a bookmark)	<p>Opens a web page where you can select whether to open the analysis with a bookmark in a web client or in an installed client.</p> <p>Example:</p> <pre>http:// myspotfireserver/  spotfire/redirect? analysis=/  Data/My%20Analysis&amp; bookmark  =7c0669f2- 9701-44d4-96b3- 7fed9e129fa</pre>	<p>In the installed client, from the <b>Bookmarks</b> popover, click <b>Copy Bookmark URL &gt; Link that Lets the Recipient Choose Client</b>.</p>	<p>When the intended audience might have access to either a web client or an installed client and you want to include a bookmark in the link.</p> <p>For example, when sharing a URL in a blog post or sending an email with the URL to people from many departments that have access to different Spotfire solutions.</p>
Link for web client users	<p>Opens the analysis in a Spotfire web client.</p> <p>Example:</p> <pre>https:// myspotfireserver/  spotfire/wp/ OpenAnalysis? file=f73c4ce4  -9174-48bb-a602- bfff3c6d862e</pre>	<p>In any client, by right-clicking an analysis in the Spotfire library section of the <b>Files and data</b> flyout and selecting <b>Copy link to item &gt; For web clients</b>.</p> <p>In the installed client, by right-clicking an analysis in the <b>Library Administrator</b>.</p> <p>In the installed client, in the <a href="#">Document Properties</a> dialog.</p>	<p>When you want the recipients of the link to open the analysis directly in a web client.</p>
-"- (with a bookmark)	<p>Can include a bookmark, if copied from the <b>Bookmarks</b> popover.</p> <p>Example:</p> <pre>https:// myspotfireserver/  spotfire/wp/ OpenAnalysis?file=/  Data/My%20Analysis&amp; bookmark  =a444c142- b411-4130-ae5- db36e5bc1ac9</pre>	<p>In any client, from the <b>Bookmarks</b> popover.</p>	-"-

Link type	Description	Retrieve where?	Use when or where?
-"- (with a configuration block)	<p>Opens the analysis in a web client with a configuration block.</p> <p>Example:</p> <pre>https:// myspotfireserver/ spotfire/wp/  OpenAnalysis?file=/ Data/My%20Analysis&amp; configurationBlock=SetFilter (columnName%3D% 22User%22%2Cvalues% 3D%7B% 22jsmith%22%7D)%3B</pre>	Manually edited.	See <a href="#">Create a configuration block</a> on the <a href="#">Community</a> for information about configuration blocks.

The Spotfire and the TIBCO Spotfire protocol handlers are registered on all Windows client machines, which means that links can be used in emails, web pages, etc., to directly open analyses. The default version of Spotfire to use when opening spotfire or tibcospotfire links can be set under [Tools > Options > Application](#) in the installed client.

A link to another analysis can be added to the text area. When an end user of the analysis clicks on the link, the link will be re-written so that the linked analysis will be opened in the installed client for persons using that client and in a web client for others, depending on context.

For information about how you can include searches in links, see [Using search expressions in shareable links](#) on page 131.

## Downloading an analysis

You can download a Spotfire analysis as a DXP file if you want to share it with a colleague by sending it in an email.

### Procedure

1. Open the analysis you want to download.
2. On the menu bar, select **File > Download as DXP file**.  
Depending on your web browser settings, the analysis is either downloaded to your computer right away, or you will be asked if you want to open or save the analysis. Save the analysis if you want to send it to someone else.

### What to do next

If you have access to the installed client, you can now open the downloaded file, or you can email it to a colleague with whom you want to share the analysis.

## Optimizing a dashboard using URL parameters

In some cases, you might need to provide an analysis to users of a small-screen mobile device. You can easily customize the analysis for the small screen, and then share the URL with small-screen users.



This technique works only if users access the analysis from a fixed URL. If the users access the analysis by navigating in the library, these settings have no effect.

If you have users who use the Spotfire Mobile App, you should avoid this optimization technique. If a user opens an analysis in the Spotfire Mobile App, the parameterized URLs will not work.

### Procedure

1. On the menu bar, select **File > Share > Copy link**.
2. In the Copy link dialog, click **Advanced**.
3. Select the check boxes for the menu options and UI elements that are appropriate to include in the analysis.



The options you selected are specified with number-logical value pairs at the end of the URL, with each number corresponding to the option and the logical value, 0, or 1, specifying whether to display the option.

4. Review and copy the URL.
5. Test the resulting analysis by opening a browser window and pasting into the address bar the copied URL.

The analysis should appear with none of the excluded display options and UI elements.



The new analysis URL does not include data selection, filters, resizing, or other manual changes that you set on any of the visualizations in the analysis before you copied and shared the URL.

# Downloads

You often have access to two different clients when using Spotfire, both the web client, and an installed client (a/k/a Spotfire Analyst). When you use the installed client as an author, you usually have access to more features for creating advanced analyses. If your administrator allows you access to the installed Spotfire client, or any additional tools, you can download the installers or components from your **My account** page on the server.

## Prerequisites

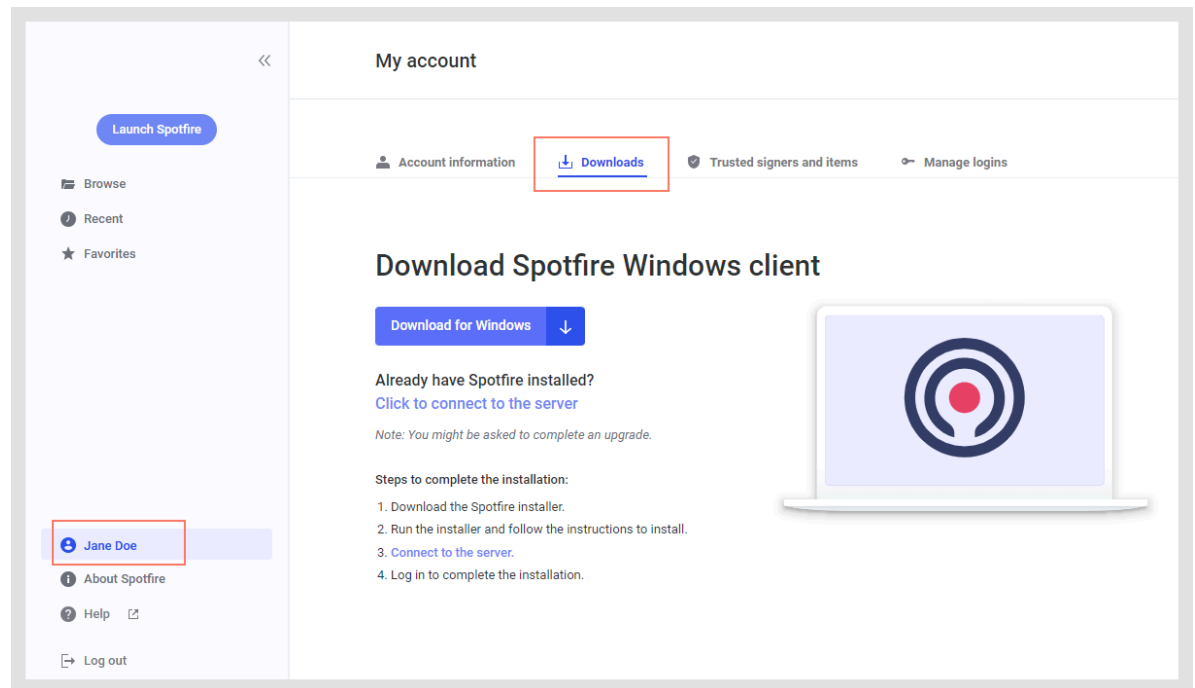
You must be logged in to your Spotfire account. You will only see downloads for those items you are allowed to use. Note that you will not get access to any more functionality than your [licenses](#) allow, even when using the installed client. If you are only allowed to open analyses in the web client, and not to create them, this will be true for the installed client as well.



If the installer package not has been deployed to your deployment area, or if the **Downloads** page has been hidden by an administrator, you will not be able to follow these instructions.

## Procedure

1. If you are working on an analysis, select **File > View library** on the menu bar.
2. In the navigation panel to the left, click on your username at the lower part of the panel.
3. In the **My account** page, click **Downloads**.



4. Follow the instructions on the **Downloads** page.

# Spotfire Documentation and Support Services

---

For information about the Spotfire® products, you can read the documentation, contact Spotfire Support, and join the Spotfire Community.

## How to Access Spotfire Documentation

Documentation for Spotfire and TIBCO products is available on the [TIBCO Product Documentation](#) website, mainly in HTML and PDF formats.

The website is updated frequently and is more current than any other documentation included with the product.

## Spotfire Documentation

The documentation for all Spotfire products is available on the [Spotfire Documentation](#) page. This page takes you directly to the latest version of each document.

To see documents for a specific Spotfire product or version, click the link of the product under 'Other versions', and on the product page, choose your version from the top right selector.

## Release Version Support

Some release versions of Spotfire products are designated as long-term support (LTS) versions. LTS versions are typically supported for up to 36 months from release. Defect corrections will typically be delivered in a new release version and as hotfixes or service packs to one or more LTS versions. See also <https://spotfi.re/lts>.

## How to Contact Support for Spotfire Products

You can contact the Support team in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the support portal at <https://spotfi.re/support>.
- For creating a Support case, you must have a valid maintenance or support contract with Cloud Software Group, Inc. You also need a user name and password to log in to <https://spotfi.re/support>. If you do not have a user name, you can request one by clicking **Register** on the website.

## System Requirements for Spotfire Products

For information about the system requirements for Spotfire products, visit <https://spotfi.re/sr>.

## How to join the Spotfire Community

The Spotfire Community is the official channel for Spotfire customers, partners, and employee subject matter experts to share and access their collective experience. The Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from Spotfire products. In addition, users can submit and vote on feature requests from within the [Ideas Portal](#). For a free registration, go to <https://spotfi.re/community>.

## Legal and Third-Party Notices

---

SOME CLOUD SOFTWARE GROUP, INC. ("CLOUD SG") SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, "INCLUDED SOFTWARE"). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

Spotfire, the Spotfire logo, TERR, and TIBCO are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries. A list of Cloud SG's trademarks and trademark guidelines is available at <https://www.cloud.com/legal>.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG's Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the Apache License, Version 2.0, which is available at <https://www.apache.org/licenses/LICENSE-2.0> and reprinted in the Addendum below.

Copyright (c) Christian Robertson / Google, Roboto font.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the "readme" file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.tibco.com/patents>.

Copyright © 1994-2024 Cloud Software Group, Inc. All Rights Reserved.



## Addendum to Legal and Third-Party Notices

Apache License  
Version 2.0, January 2004  
<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
  - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
  - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
  - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
  - (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions

for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.
6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

# Index

---

## A

Add calculated columns [730](#)  
 Add columns [187](#), [189](#)  
 Add connection [100](#), [104](#)  
 Add map chart [392](#)  
 Add relations [202](#)  
 Add rows [182](#), [184](#)  
 Add view to connection [197](#)  
 Aggregated values [421](#), [514](#)  
 Alphabetical sorting [156](#)  
 Arrange visualizations [603](#), [604](#)

## B

Bookmarks [698](#)  
 Broken column match [184](#), [189](#)  
 Broken transformations [194](#)

## C

Calculated column [730](#)  
 Calculated columns [730](#)  
 Categorization of columns [140](#)  
 Change a value [239](#)  
 Change categorization [140](#)  
 Change column description [158](#)  
 Change column name [158](#)  
 Change data table name [181](#)  
 Change data type [158](#)  
 Change display name [523](#)  
 Change formatting [142](#)  
 Change sort order [156](#)  
 Column description [158](#)  
 Column matching [184](#), [189](#)  
 Column name [158](#)  
 Column view [136](#)  
 Combine columns [174](#)  
 Comments in expressions [846](#)  
 Configure map [396](#), [414](#), [420](#), [421](#), [423](#)  
 Configuring on-demand data [61](#)  
 Copy coordinates [418](#)  
 Custom format [142](#)  
 Custom sort order [156](#)

## D

Data canvas [178](#)  
 Data format [174](#)  
 Data functions [662](#), [664](#), [666](#), [668](#), [670](#), [671](#), [671](#), [672](#), [672](#), [675](#)  
 Data in analysis flyout [135](#)  
 Data loading [193](#), [227](#), [228](#)

Data panel [135](#)  
 Data preparation [136](#)  
 Data source [195](#), [233](#)  
 Data table name [181](#)  
 Data table overview [178](#)  
 Data type [158](#)  
 Data wrangling [136](#)  
 Decimals [61](#), [142](#)  
 Default sort order [156](#)  
 Delete a column [168](#)  
 Delete a data table [182](#)  
 Deleting items from library [960](#)  
 Details on selected column [136](#)  
 Display name [523](#)

## E

Edit added columns [189](#)  
 Edit added rows [184](#)  
 Edit connection [197](#)  
 Edit map [396](#)  
 Empty values [237](#)  
 EPSG [413](#)  
 Exclamation mark in source view [184](#)  
 Expanded data panel [136](#)  
 Expressions [846](#)

## F

Feature layer [399](#)  
 Formatting [142](#), [147](#), [148](#), [150](#)

## G

Geocoding [169](#), [420](#)  
 Geocoding tables [172](#)  
 Geographic coordinates [418](#)  
 Geographic location and geocoding [169](#)  
 Geographical context [392](#)  
 Google Analytics [100](#), [197](#)

## H

Hide optional parameters [666](#)

## I

Information about source [178](#)  
 Insert calculated column [730](#)  
 Insert rows [182](#), [187](#)

**J**

Join settings [189](#)

**K**

Key columns [199](#)

**L**

Layout [603](#), [604](#)  
 Line connection in maps [423](#)  
 Location [418](#)  
 Lock visualization area [623](#)

**M**

Map chart [392](#)  
 Map chart layer [393](#), [399](#), [411](#), [413](#)  
 Map chart properties [393](#)  
 Map layer [410](#)  
 Map navigation [419](#)  
 Map shapes [399](#)  
 Map sublayer [414](#)  
 Marker layer [393](#), [423](#)  
 Marking in maps [419](#)  
 Matching columns [184](#), [189](#)  
 Modify views [197](#)

**N**

Natural sorting [156](#)  
 Number of decimals [142](#)

**O**

On-demand [61](#)  
 Order of values [156](#)

**P**

Parameterizing [993](#)  
 Primary key [199](#)

**R**

Reapply markings [532](#)  
 Recommendations based on selected data [136](#)  
 Refresh data [227](#), [228](#)  
 Reload data [227](#), [228](#)  
 Reload data table [178](#)  
 Remove a column [168](#)  
 Remove a data table [182](#)  
 Remove data source [195](#)  
 Remove operations in the data canvas [194](#)  
 Remove transformation [194](#)  
 Remove view from connection [197](#)

Rename an axis [523](#)  
 Rename column [158](#)  
 Rename data table [181](#)  
 Replace data source [233](#)  
 Replace empty values [237](#)  
 Replace value [239](#), [240](#), [241](#), [242](#)  
 Rotate markers [396](#)

**S**

Salesforce [104](#), [197](#)  
 Save data [193](#)  
 Saving an analysis [959](#)  
 Short/wide [174](#)  
 Showing direction [396](#)  
 Sort order [156](#)  
 Source data [197](#)  
 Source view [178](#)  
 Spelling error [239](#)  
 Split column [165](#)  
 Standard alphabetical sorting [156](#)  
 Store data [193](#)

**T**

Tall/skinny [174](#)  
 The f(x) flyout [675](#)  
 This is Spotfire [13](#)  
 Tile map service [411](#)  
 Time differences [752](#)  
 TMS [411](#)  
 Tuning of data functions [672](#)

**U**

Unpivot [174](#)  
 URL [993](#)

**W**

Web map service [413](#), [414](#)  
 WMS [413](#)  
 WMS properties [414](#)

**Z**

Zooming in maps [419](#)